



**Escola Politècnica Superior  
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# **TRABAJO DE FINAL DE CARRERA**

**TITULO DEL TFC:** Integración de Farsite en el entorno de simulación para UAS.

**TITULACIÓN:** Ingeniería Técnica de Telecomunicación, especialidad Telemática

**AUTOR:** M<sup>a</sup> Victoria Noguero Planas

**DIRECTOR:** Cristina Barrado Muxí

**FECHA:** 12 de Mayo de 2010

**Título:** Integración de Farsite en el entorno de simulación para UAS.

**Autor:** M<sup>a</sup> Victoria Noguero Planas

**Director:** Cristina Barrado Muxí

**Fecha:** 12 de Mayo de 2010

## **Resumen**

Actualmente los incendios son una de las causas en la pérdida y destrucción del medioambiente. Por esta razón son muchos los profesionales que trabajan para prevenir y/o extinguirlos, ya que de esta manera la pérdida no es tan considerable. Es por lo que este proyecto de final de carrera presenta un servicio para facilitar el trabajo que realiza el cuerpo de bomberos en su prevención y extinción de incendios.

El servicio ha sido diseñado para ser utilizado como una pieza más en un UAV, avión no tripulado que vuela por sí solo sin necesidad de que nadie se encuentre en su interior para poderlo pilotar.

La idea principal del proyecto es hacer volar un UAV sobre un incendio en una zona particular y utilizar dicho servicio para que informe de la temperatura que hay en las coordenadas por las que va pasando el avión.

Principalmente para realizar el proyecto se ha hecho uso del simulador de incendios Farsite que es el que aporta la información de la temperatura del fuego. Utilizando el simulador de vuelo Flight Gear se simula la función del UAV que sobrevolará el incendio gracias a una ruta de vuelo previamente establecida. Se ha implementado un programa en C# que es el encargado de recibir las coordenadas GPS por las que va pasando el avión e informar de la temperatura exacta en la que se encuentra el suelo correspondiente a dicha coordenada.

Para que los simuladores y el programa nombrados anteriormente se puedan comunicar entre sí, se usa el middleware MAREA que ha sido desarrollado por el equipo de investigación ICARUS de la Escuela Politécnica Superior de Castelldefels.

Todo este sistema forma parte del simulador Icarus Simulation Integrated Scenario (ISIS) que es el responsable de simular que todos los servicios utilizados se encuentren dentro del UAV.

**Title:** Integración de Farsite en el entorno de simulación para UAS.

**Author:** M<sup>a</sup> Victoria Noguero Planas

**Director:** Cristina Barrado Muxí

**Date:** May, 12th 2010

## Overview

Nowadays one of the most significant causes of environment destruction is the fire. Due to that reason, a lot of work to prevent and exterminate it. Consequently this final career project presents a service to make easier the work of firefighters in prevention and extermination fire. This service has been designed to use as piece in UAV (Unmanned Aerial Vehicle) without human pilot.

The main idea of this project is to fly the plane around the fire and send the coordinates to the program, and then the program calculates the temperature at this coordinate and shows it in the screen.

To do the project the Farsite has been used simulation fire; which contains the temperature values. Flight Gear simulates the UAV and its function is to fly around the fire with a preprogrammed flight path. To receive the coordinates of the UAV and the temperature at that point a program based in C# has been used.

To be able to communicate the simulators and the C# program between themselves, the Marea middleware has been used. This middleware has been developed by ICARUS investigation team of "Escuela Politécnica Superior de Castelldefels".

All the system is a piece of the Icarus Simulation Integrated Scenario (ISIS), it is responsible to simulate all services used in the plane.

# ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1. CONOCIMIENTOS PREVIOS.....</b>	<b>3</b>
1.1. <b>Sistemas de Información Geográfica .....</b>	<b>3</b>
1.1.1 Raster .....	4
1.1.2 Vectorial.....	5
1.2. <b>Sistemas de Coordenadas.....</b>	<b>5</b>
1.2.1 Global Position System (GPS) .....	6
1.3. <b>Proyecciones .....</b>	<b>7</b>
1.2.2 Universal Transversal de Mercator (UTM) .....	7
1.2.3 Proyección Lambert azimuthal equal-area .....	10
<b>CAPÍTULO 2. FARSITE.....</b>	<b>11</b>
2.1 <b>Concepto y uso del programa.....</b>	<b>11</b>
2.2 <b>Datos de entrada y salida .....</b>	<b>12</b>
2.2.1 Datos de entrada .....	12
2.2.2 Datos de salida .....	14
2.3 <b>Estructura de la información.....</b>	<b>15</b>
2.1.1 Ficheros de entrada.....	15
2.1.2 Parámetros de entrada.....	19
2.1.3 Ficheros de salida .....	21
<b>CAPÍTULO 3. GENERACIÓN DE LOS RASTERS PARA UN INCENDIO EN CATALUÑA NORTE .....</b>	<b>22</b>
3.1 <b>Creación de los rasters de entrada para el simulador Farsite.....</b>	<b>22</b>
3.1.1 Global Mapper .....	23
3.1.2 Realización del Script para la creación de los rasters <i>canopy</i> y <i>fuel</i> .....	28
3.2 <b>Pruebas.....</b>	<b>30</b>
3.2.1 Prueba 1: Efectos del viento.....	30
3.2.2 Prueba 2: Efecto del tipo de combustible .....	31
3.2.3 Prueba 3: Efecto de la orografía .....	32
<b>CAPÍTULO 4. EL USO DE UAS EN INCENDIOS .....</b>	<b>34</b>
4.1 <b>Servicios diseñados para los puntos calientes .....</b>	<b>34</b>
4.2 <b>Marea .....</b>	<b>34</b>
4.3 <b>Icarus Simulation Integrated Scenario .....</b>	<b>37</b>
4.3.1 FarsiteSim.....	40
4.3.2 IRService .....	40
4.3.3 Fire Perimeter Visor.....	45

4.4	Demostración.....	45
<b>CAPÍTULO 5. CONCLUSIONES .....</b>		<b>49</b>
5.1	Conclusiones Tecnológicas .....	49
5.2	Conclusiones personales .....	50
5.3	Futuras líneas de trabajo .....	50
5.4	Impacto Medioambiental.....	51
<b>BIBLIOGRAFIA .....</b>		<b>52</b>
<b>ANNEXOS.....</b>		<b>55</b>
1.1.	ANNEXO 1: MODELOS DE COMBUSTIBLE PROPUESTOS POR ROTHERMEL PARA EL NFDRS Y ADAPTADOS POR LA DGCN A ESPAÑA.....	56
1.2.	ANNEXO 2: SCRIPT .....	58
1.3.	ANNEXO 1: RASTERS DE ENTRADA PARA FARSITE .....	60



## INTRODUCCIÓN

Los incendios forestales son un gran problema hoy en día, debido a que en la actualidad las zonas boscosas escasean, provocado en parte por la proliferación de construcciones que ha habido en los últimos años.

Por lo tanto es indispensable proteger al máximo las zonas verdes y forestales, ya que si no hacemos las actuaciones pertinentes y necesarias poco a poco provocaremos la desaparición paulatina de las mismas.

El hecho de que haya un incendio, no solo repercute en la zona quemada, sino que además provoca muchas más consecuencias ambientales: como la erosión hídrica (cuando el agua se lleva la tierra), la emisión de dióxidos de carbono en la atmósfera, la desaparición de medios de hábitat, la destrucción e incluso desaparición de la flora autóctona del lugar (dependiendo de la cantidad de zona quemada y el tipo de incendio que haya sido), y un sinfín de consecuencias más.

Una consecuencia mucho más visual e impactante que las comentadas anteriormente es la pérdida de valor paisajístico en una zona determinada. Cuando se ha producido un incendio la zona quemada queda desértica y carbonizada.

Los incendios también provocan una gran repercusión tanto en la población local como en su economía. Son muchas las personas que por causa de los incendios son perjudicadas económicamente por culpa de la pérdida de sus bienes: terrenos agrícolas, ganadería, viviendas, etc., y que como consecuencia provoca la pérdida de sus ingresos.

Además los incendios provocan muchas situaciones dramáticas si las personas que se ven involucrados en ellos sufren daños físicos, psíquicos, o incluso lleguen a pagarlo con sus vidas.

A parte de las personas que de forma involuntaria se han visto afectadas e involucradas en los incendios, también se debe tener en cuenta la gente que participa en su extinción (bomberos, voluntarios, etc.) que lamentablemente algunas veces también son víctimas de ellos.

Es por ello que este proyecto tiene como objetivo ser una herramienta útil para la prevención y la extinción de los incendios.

Para la realización de este proyecto se han utilizado tres simuladores:

- Simulador de incendios mediante el simulador Farsite [1].
- Simulador de escenario ISIS (*Icarus Simulation Integrated Scenario*).
- Simulador de un avión no tripulado (UAV, *Unmanned Aerial Vehicle*) mediante el simulador Flight Gear [2].

Se ha hecho uso de los Sistemas de Información Geografica ( tecnología SIG) para poder procesar los datos geográficos que necesita el sistema.

También se ha implementado un programa en C# nombrado IRService. Éste recoge la información de salida que genera el simulador Farsite e interactúa con el simulador Flight Gear a través del ISIS.

Para que los simuladores y el programa puedan comunicarse entre ellos se ha utilizado el middleware MAREA .

El objetivo final de este proyecto es:

- Cuando en una zona de Catalunya haya un incendio, un avión no tripulado sobrevolará dicha zona.
- El avión estará equipado con una cámara infrarroja que enviará al IRService las coordenadas GPS de los puntos por donde vuele.
- El programa escaneará la información que se recibe de la cámara infrarroja y cuando detecte que en algún punto por donde pasa el avión hay fuego, enviará un aviso con la información del punto exacto y la temperatura en dicho punto.

La implementación y realización de la cámara térmica ya se ha llevado a cabo en otro TFC de la EPSC titulado “Red-Eye. Sistema para la monitorización semiautomática de incendios forestales” [3].

Pero como todo proyecto para conseguir el objetivo final se tienen que realizar un conjunto de pruebas mediante simuladores que reproduzcan las funciones anteriormente nombradas. Para ello el proyecto ha substituido la cámara infrarroja por el simulador Farsite y el avión no tripulado por el simulador Flight Gear. De esta forma la idea principal del proyecto es:

- Simular un incendio en una zona determinada de Catalunya con el simulador Farsite.
- El avión del simulador Flight Gear ira volando por la misma área que el simulador Farsite simula y ira enviando las coordenadas GPS de su posición al programa IRService
- El programa recibe las coordenadas GPS del simulador Flight Gear y hace un escaneo de todos los datos de salida que genera el simulador Farsite .
- Cuando el IRService detecte que hay fuego en una de las coordenadas que le ha enviado el simulador Flight Gear, el programa enviara un aviso con la información de la coordenada exacta y la temperatura en dicho punto.



# CAPÍTULO 1. CONOCIMIENTOS PREVIOS

## 1.1. Sistemas de Información Geográfica

Los Sistemas de Información Geográfica (SIG) se encuentran dentro de la familia de los Sistemas de Información.

Los Sistemas de Información no son más que programas o conjuntos de programas diseñados para representar y gestionar grandes volúmenes de datos sobre ciertos aspectos del mundo real. De esta manera, un SIG es un Sistema de Información diseñado para trabajar con datos georreferenciados mediante coordenadas espaciales o geográficas.

Desde otro punto de vista se trata de una extensión del concepto de base de datos: un SIG es una base de datos computerizada que contiene información espacial. En un SIG se almacena información cartográfica (con lo que es posible conocer la localización exacta de cada elemento en el espacio y con respecto a otros elementos) e información alfanumérica (datos sobre las características o atributos de cada elemento geográfico). Este hecho, el trabajar con información espacial, es lo que diferencia básicamente a los SIG de los Sistemas de Información [4].

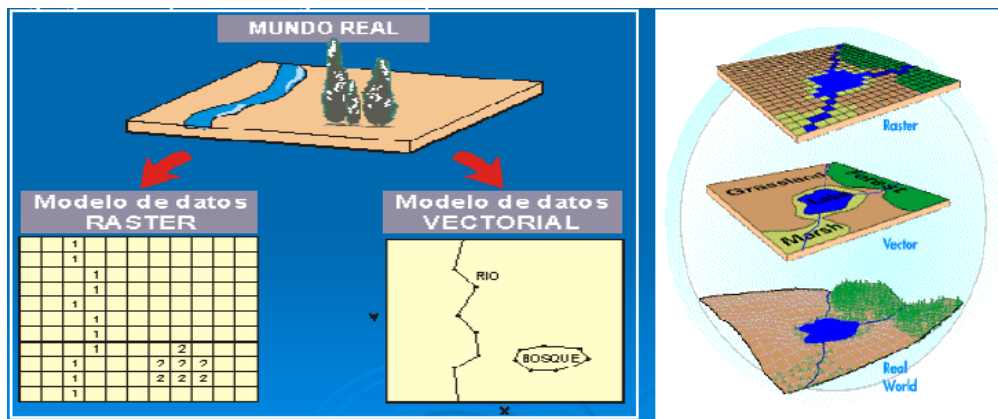
En un sentido más genérico, los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones [5], como por ejemplo los programas ArcGIS, OpenGIS, gvSIG, capaware, etc.

Una de las ventajas de utilizar SIG es que el sistema permite descomponer la realidad en distintos temas, es decir, en distintas capas o estratos de información de la zona que se desea estudiar. De esta manera el analista puede trabajar sobre cualquiera de esas capas según las necesidades del momento y puede relacionar las distintas capas entre sí, lo que concede a estos sistemas unas sorprendentes capacidades de análisis [6].

La información que contienen los SIG representan cada uno de los elementos que existen en la vida real, ya sean carreteras, ríos, casas, etc.

Pero, como se almacena la información?

El SIG dispone de dos modelos de datos para almacenar de forma eficaz la información necesaria: modelo de datos raster y modelo de datos vectorial. En la **Fig 1.1** se puede observar como para representar el mundo real se utiliza cada uno de los modelos nombrados.



**Fig. 1.1** Representación de los dos modelos de almacenamiento de datos [7].

### 1.1.1 Raster

Este tipo de modelo de datos es utilizado habitualmente para estudios medioambientales. Ya que es especialmente útil cuando se tienen que describir objetos geográficos con límites difusos, como por ejemplo la dispersión de una nube contaminante, donde los contornos no son absolutamente nítidos.

Los SIG rasters dividen la zona de afectación de la base de datos en una retícula o malla regular de pequeñas celdas (denominadas píxeles) y atribuyen un valor numérico a cada celda para que de esta manera se pueda representar de forma digital [7], tal y como se puede observar en la **Fig.1.2**.

Debido a que la malla es regular y que la posición del centro de cada retícula es conocida, todos los píxeles están georreferenciados. Cada malla contiene los píxeles ordenados en filas y columnas. Este número de filas y columnas es el que definirá la resolución de dicho raster.



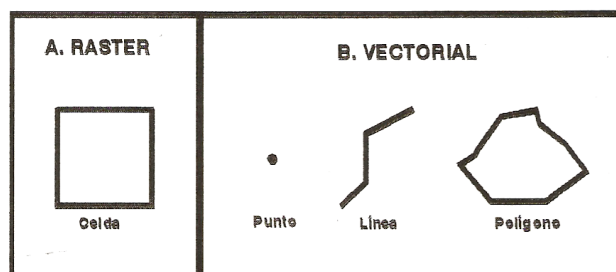
**Fig. 1.2** Organización de la información en el modelo de datos raster [7].

El modelo de datos raster puede tener diferentes tipos de formatos, los que más se utilizan son: ASCII, BLOB, TIFF, etc.

### 1.1.2 Vectorial

Este modelo de datos utiliza vectores para la descripción de los objetos geográficos. Estos vectores son definidos por pares de coordenadas relativas a algún sistema cartográfico como los que se van a describir en el siguiente apartado.

Las propiedades constituyen el criterio de diferenciación de los individuos geográficos que existen en el mundo real: las entidades. Estas entidades son representadas por medio de objetos en la base de datos. Así por ejemplo un lago (la entidad) puede representarse en un SIG mediante un polígono (el objeto), tal y como se puede observar en la **Fig.1.3**. En general, este modelo es adecuado para cuando se trabaja con objetos geográficos con límites bien establecidos, como pueden ser fincas, carreteras, etc.



**Fig. 1.3** Tipos básicos de objetos espaciales: puntos, líneas, polígonos y celdas.

El modelo de datos vectorial puede tener diferentes tipos de formatos, los que más se utilizan son: AI, WMF, KML, etc.

En este proyecto se ha utilizado el SIG del tipo raster para que el simulador Farsite pueda simular el incendio en la zona deseada.

Antes de analizar los datos en el SIG la cartografía (elaboración de mapas geográficos) debe estar toda ella en una misma proyección y sistema de coordenadas. Para ello, a continuación se van a describir las diferentes proyecciones y sistemas de coordenadas que se han utilizado para llevar a cabo este proyecto.

## 1.2. Sistemas de Coordenadas

La tierra se compone de puntos donde cada uno de ellos identifica un lugar diferente, pero para poder diferenciar unos puntos de otros se les tiene que asignar un nombre. De esta manera un sistema de coordenadas es el sistema que le da el nombre a cada uno de estos puntos para poder identificarlos de

forma individual. En un sistema de coordenadas proyectadas, los puntos se identifican por las coordenadas x,y en una malla.

En este apartado se explicará la coordenada GPS, coordenada que utiliza el simulador de vuelo utilizado en este proyecto para especificar su posición.

### 1.2.1 Global Position System (GPS)

El GPS (Global Position System) es un sistema de posicionamiento por satélite que permite estimar de forma precisa la posición tridimensional para usuarios equipados con receptores adecuados. La posición la calculan los receptores GPS gracias a la información recibida desde satélites en órbita alrededor de la Tierra [5]. Consiste en una red de 24 satélites, propiedad del Gobierno de los Estados Unidos de América y gestionada por el Departamento de Defensa, que proporciona un servicio de posicionamiento para cualquier lugar del planeta y a cualquier hora, independientemente de las condiciones atmosféricas [8].

Según comenta el artículo de Jorge Blanco [9], cada uno de estos 24 satélites transmiten dos tramas de código distintas basadas en una serie de parámetros Keplerianos denominadas Almanaque y Efemérides. Estos parámetros son fundamentales en el proceso de funcionamiento del sistema, ya que informan sobre la situación del satélite en la órbita, la operatividad del mismo y los datos de fecha y hora. Estas tramas de código no son generales al sistema, sino particulares de cada satélite, por lo que cada uno de estos dispositivos incorpora un código de identificación.

Los satélites transmiten continuamente su situación orbital y la hora exacta, hecho que aprovecha el receptor para calcular su distancia a los mismos, basándose en el tiempo transcurrido entre la emisión de los satélites y lo que tarda en recibir esa señal mediante unas sencillas operaciones aritméticas.

Si el receptor consigue captar las señales de al menos 3 satélites (como se puede observar en la **Fig.1.4**), será capaz de determinar la posición relativa que ocupa en la superficie terrestre es decir la latitud y la longitud (coordenadas 2D que pueden venir expresadas en cualquier tipo de unidad de medida válida, según el uso que se les vaya a dar). Si además el GPS también capta la señal de algún satélite adicional, éste será capaz de determinar la posición en 3D, añadiendo la coordenada de la altura con respecto al nivel del mar a las ya conocidas longitud y latitud.

En este proyecto las coordenadas GPS las utiliza el simulador Flight Gear.



**Fig. 1.4** Representación de un receptor GPS captando la señal de 3 satélites.

### 1.3. Proyecciones

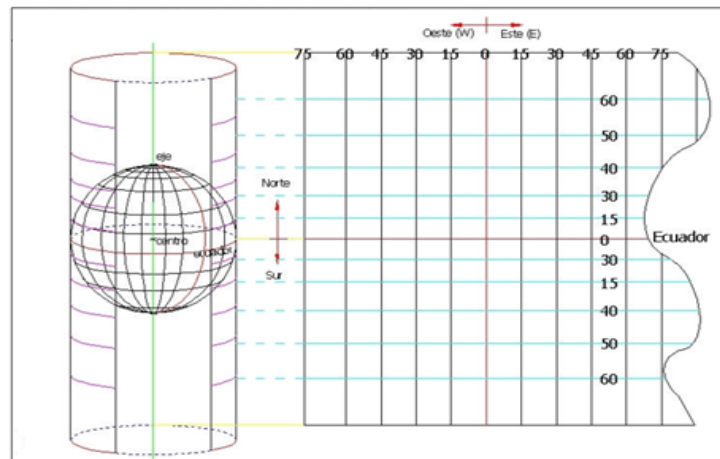
Las proyecciones se utilizan cuando no se puede representar un objeto de forma real en un plano, como por ejemplo pasa en la representación de la tierra. Al ser la tierra “redonda” es difícil que se pueda representar en un plano con su forma real, para ello se han creado las proyecciones cartográficas.

La proyección cartográfica es un sistema de representación gráfico que establece una relación ordenada entre los puntos de la superficie curva de la Tierra y los de una superficie plana (mapa). Estos puntos se localizan auxiliándose en una red de meridianos y paralelos, en forma de malla. La única forma de evitar las distorsiones de esta proyección sería usando un mapa esférico pero, en la mayoría de los casos, sería demasiado grande para que resultase útil [10].

En este apartado se van a explicar el tipo de proyecciones utilizadas para realizar el proyecto: *Universal Transversal de Mercator* (UTM), *Lambert azimuthal equal-area*.

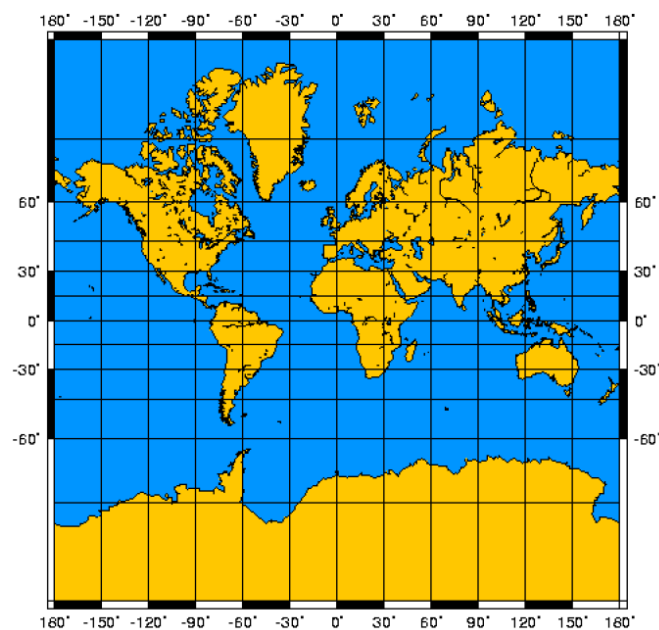
#### 1.2.2 Universal Transversal de Mercator (UTM)

El sistema UTM fue desarrollado por el Cuerpo de Ingenieros del Ejército de los Estados Unidos en la década de 1940. El sistema se basó en un modelo elipsoidal de la Tierra. La "proyección transversa de Mercator" es una variante de la "proyección de Mercator" que fue desarrollada por el geógrafo flamenco Gerardus Mercator en 1569 [11]. Éste sistema está dentro de las llamadas proyecciones cilíndricas debido a que emplea un cilindro situado de forma tangente al elipsoide en el ecuador para proyectar las situaciones geográficas [12] tal y como se puede observar en la **Fig.1.5**.



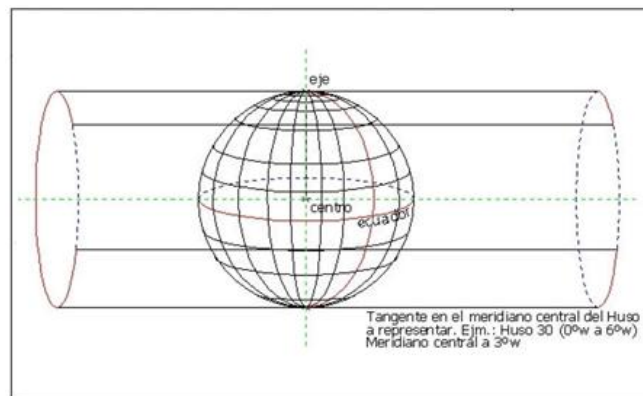
**Fig. 1.5** Proyección mercator.

Ésta, como en toda proyección cartográfica, cuando se intenta ajustar una superficie curva en una superficie plana, la forma del mapa es una distorsión de la verdadera configuración de la superficie terrestre. La proyección de Mercator va exagerando el tamaño y distorsionando las formas a medida que nos alejamos de la línea del ecuador. Si se observa la **Fig.1.6** se puede ver de forma clara lo comentado anteriormente, las cuadrículas que se encuentran a los extremos de la figura son de un tamaño mayor que las que se pueden ver en el centro de ella. Es por esto que aunque la proyección de Mercator es todavía muy usada en navegación, los críticos argumentan que no es indicada para representar el mundo completo dada la distorsión de las áreas.



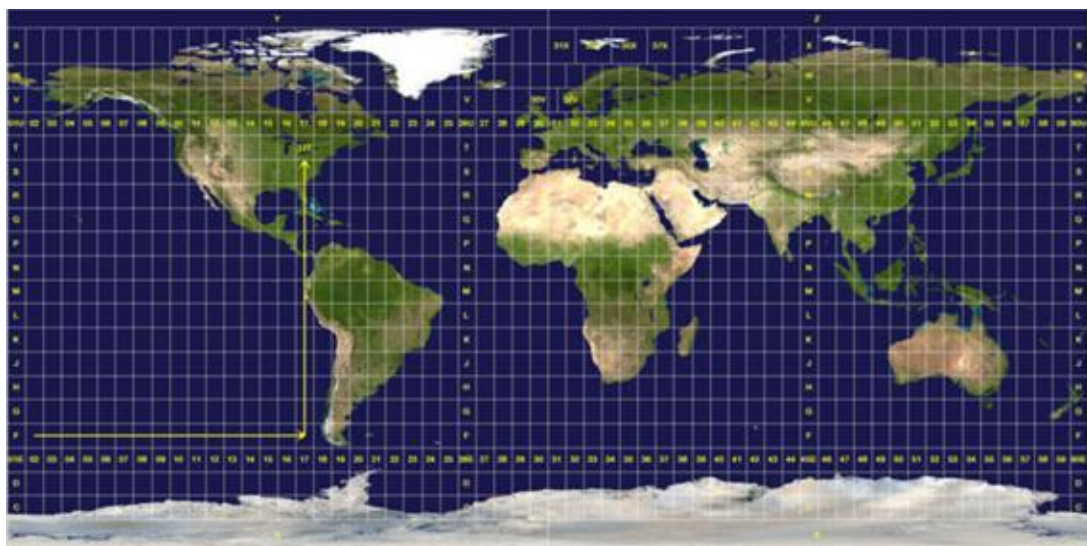
**Fig. 1.6** Globo terráqueo en la proyección mercator.

Como solución al problema, actualmente se usa la proyección UTM que como ya se ha comentado anteriormente toma como base la proyección Mercator. En este caso la posición de cilindro de proyección es transversal respecto del eje de la tierra (ver **Fig.1.7**). De esta manera tiene la ventaja de que ningún punto está alejado del meridiano central de su zona, por lo que las distorsiones son pequeñas. Los meridianos se proyectan sobre el plano con una equidistancia entre ellos. Sin embargo los paralelos se van separando a medida que nos alejamos del Ecuador, por lo que al llegar al polo las deformaciones serán infinitas. Es por ello que solo se representa la región entre los paralelos 84°N y 80°S. de esta manera la esfera se representa en trozos, no entera. Para ello se divide la Tierra en husos de 6° de longitud cada uno.



**Fig. 1.7** Proyección UTM.

Como se puede observar en la **Fig.1.8** en este caso las cuadrículas son todas uniformes y el error de proyección es más pequeño. En esta figura sí que se representan los polos, pero si se observa la cuadrícula que les corresponde tiene una forma más alargada que las demás, por esto se corta en un cierto límite, ya que si no tendería a infinito.



**Fig. 1.8** Husos y zonas UTM.



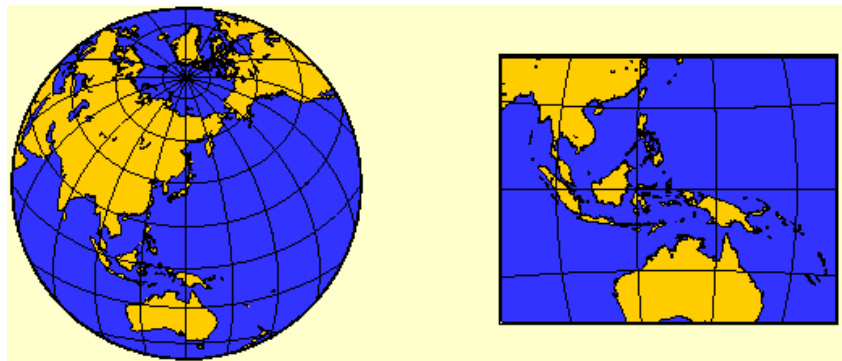
Para este proyecto son necesarias las coordenadas UTM para que el simulador Farsite indique el punto exacto en el que se encuentra el incendio.

### 1.2.3 Proyección Lambert azimuthal equal-area

Esta proyección fue creada por Lambert en 1772, y se usa típicamente para representar grandes regiones del tamaño de continentes y hemisferios.

Se representa con exactitud el área en todas las regiones de la esfera, pero no representan con exactitud los ángulos [13]. Ópticamente da una imagen plana de la esfera en la que cada región aparece con el área correcta pero quizás de una forma distorsionada, tal y como se puede apreciar en la **Fig.1.9**.

Las áreas representadas coinciden con las reales. La distorsión es cero en el centro de la proyección para cada plano que se represente, pero esta distorsión aumenta radialmente conforme se aleja del centro [14].



**Fig. 1.9** Proyección Lambert azimuthal equal-area.

En este caso la proyección Lambert azimuthal equal-area es la que utilizan los mapas rasters originales que se obtienen para poder simular el incendio..



## CAPÍTULO 2. FARSITE

### 2.1 Concepto y uso del programa

Farsite es un software que permite hacer un estudio del comportamiento y crecimiento del fuego en un lugar determinado. Es útil para avanzarse a los movimientos que podría tener un incendio y de esta manera prevenir como extinguirlo de forma efectiva y rápida.

Debido a esto, son muchos los profesionales relacionados en el sector de la prevención, extinción y estudio del comportamiento de los incendios que utilizan este programa. Cada organización o empresa que se dedica a ello puede obtener de Farsite una utilidad dentro de su propio campo de trabajo.

La Dirección General de Emergencias y Seguridad Civil de Cataluña es un claro ejemplo de organización que utiliza el programa Farsite como una herramienta útil para el desempeño de su labor.

A través de la red se pueden encontrar algunos artículos, presentaciones y trabajos, donde en cada uno de ellos se explica el papel que ha tenido Farsite en su estudio.

La presentación de Moisés Galán y Domingo Molina [15], trata sobre el estudio de la prevención de grandes incendios en el Berguedá. Utiliza el programa Farsite para simular la estrategia que querían llevar a cabo en la realidad: separar 2 macizos forestales y así crear un cortafuego. Gracias a la simulación de Farsite y todo el estudio que han realizado para que la simulación sea lo más real posible, han podido verificar el resultado.

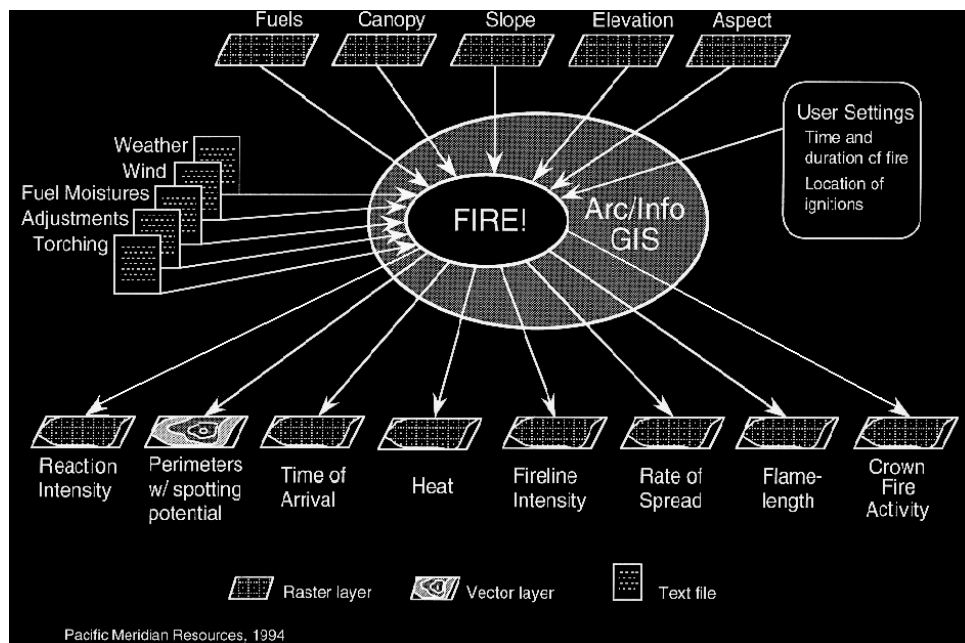
En el artículo “Pre-extinción de grandes incendios forestales : El caso de Rialb” [16], realizaron numerosas simulaciones con el programa donde en cada una de ellas utilizaban diferentes métodos para la extinción de un posible fuego. Finalmente como conclusión de su estudio han podido diseñar las infraestructuras necesarias desde el punto de vista de la extinción pasiva (pre-extinción) y activa de incendios forestales.

Por otro lado el Instituto Técnico de Canarias junto con tres departamentos de la Universidad de las Palmas de Gran Canaria [17], han llevado a cabo un proyecto en el que se hace uso del simulador de Farsite. El proyecto consiste en el desarrollo de una herramienta software para la Gestión Virtual de Emergencias que como Farsite, su finalidad es simular y prevenir el comportamiento del fuego en las islas Canarias. La diferencia está en que este nuevo software, utiliza diferentes servidores y programas que entre ellos hacen que la introducción de parámetros para la simulación sea mucho más rápida y los resultados de la simulación se muestren en 3D.

En todas las simulaciones realizadas en los estudios, trabajos y presentaciones sobre Farsite, se han tenido que introducir en el programa los ficheros de

entrada necesarios y obligatorios para su simulación. De la misma manera el programa al iniciarse y finalizar la simulación, genera ficheros de salida con los que se puede llevar a cabo el estudio del incendio.

En la **Fig.2.1** se pueden ver todos los principales rasters y ficheros de entrada y salida del simulador Farsite. En la parte superior se observan los rasters de entrada necesarios para su simulación, en la parte de la izquierda se encuentran los ficheros de texto de entrada, a la derecha indica que se le tienen que entrar diferentes parámetros de configuración, y por ultimo en la parte de abajo se pueden observar los diferentes rasters de salida que farsite tiene la opción de generar.



**Fig. 2.1** Datos que utiliza y/o genera el programa Farsite [18].

## 2.2 Datos de entrada y salida

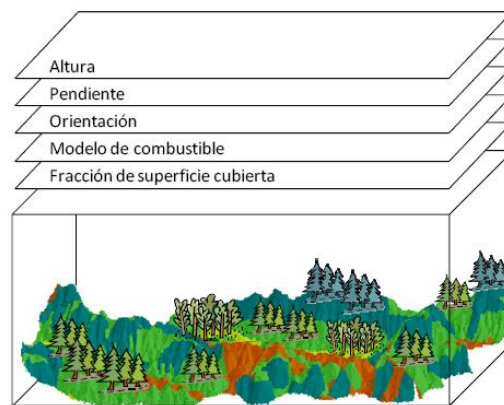
### 2.2.1 Datos de entrada

Para llevar a cabo la simulación, Farsite debe disponer de un conjunto de parámetros que le son indispensables para poderla realizar. Estos parámetros son los que informan del nivel de temperatura, territorio, tiempo y vegetación que se tiene en un determinado momento la zona a estudiar.

Los parámetros que aportan información sobre la topografía y el combustible de la zona son datos cartográficos SIG, que deben de estar en formato raster, los cuales se trata simplemente de diferentes capas de información.

Para que la simulación se pueda iniciar correctamente es obligatorio que se introduzcan 5 de las 10 capas que Farsite permite utilizar para su simulación. Dichas capas necesarias son las siguientes: altura (*elevation*), pendiente (*slope*), orientación (*aspect*), modelo de combustible (*fuel*) y fracción de superficie cubierta (*canopy*), altura a la base de copas (*standHeight*), altura media de los arboles (*crown base height*), densidad aparente de copas (*crown bulk density*), carga inútil (*duff loading*), leña gruesa (*coarse woody*).

En la **Fig.2.2**, se puede observar las 5 capas obligatorias que Farsite necesita para su adecuado funcionamiento.



**Fig. 2.2** Capas de entrada necesarias para la simulación de Farsite [19].

Estas capas una vez introducidas en el programa se pueden visualizar en él de forma independiente.

Los factores tanto meteorológicos (ejemplo: la dirección e intensidad del viento), como del tipo de vegetación que hay en la zona, se requieren tener en formato ASCII. Éstos también se pueden visualizar en el programa, y/o editarlos mediante un editor de texto [20].

En la siguiente tabla (**Tabla 1.1**), se pueden observar todos los rasters y ficheros de entrada que se le pueden introducir a Farsite.

**Tabla 1.1** Información de entrada para el programa Farsite**a) Paisaje****b) Ambiental**

Capas Raster Requeridas	Tipología de archivo	Datos
Elevación Pendiente Orientación Modelo de combustible Cobertura de copas	Meteorológico	Precipitación. Temperaturas máximas y mínimas. Humedades relativas máximas y mínimas.
Altura a la base de las copas <sup>1</sup> Altura media de los árboles <sup>1</sup> Densidad aparente de copas <sup>1</sup> Carga útil <sup>1</sup> Leña gruesa <sup>1</sup>	Viento	Dirección del viento. Velocidad del viento. Porcentaje de nubosidad.
	Contenido de humedad	Combustibles muertos. Combustibles vivos.

<sup>1</sup>Capas SIG opcionales

Con todos estos parámetros a tener en cuenta, lo que se consigue, es que la simulación sea mucho más parecida a la realidad, ya que de esta manera el programa puede calcular de forma más exacta la rapidez con la que se extiende el incendio y la creación de nuevos focos secundarios.

Incluso para que la simulación sea más real y exacta, Farsite permite introducir diferentes elementos que actúan como barreras en un incendio (cortafuegos, ríos, etc.). De esta manera también permite avanzarse al comportamiento del incendio si se decide hacer uso de unos cortafuegos para intentar extinguirlo.

### 2.2.2 Datos de salida

Una vez que la simulación está en marcha el programa ira generando una serie de ficheros de salida, y éstos se irán modificando a medida que la simulación vaya avanzando. Mientras la simulación está en curso se puede observar de forma gráfica como va transcurriendo el incendio.

Por otra parte también se pueden generar hasta ocho nuevos ficheros rasters de salida, donde cada uno de ellos aporta una nueva información sobre la simulación.

Los ocho nuevos rasters son: tiempo de llegada (TOA, *time of arrival*), altura de llama( FML, *flame lenght*), velocidad de propagación, (ROS, *rate of spread*), dirección de propagación(SDR, *spread direction*), intensidad del frente de llama(FLI, *firelin intensity*), calor por unidad de area (HPA, *head for area*), energia de la combustión (RCI, *reaction intensity*), actividad de fuego en copas (CFR, *crown fire activity*).

Estos ficheros se deben seleccionar previamente al inicio de la simulación. Si no se selecciona ninguno de ellos, éstos no se generarán. De esta forma se escoge el tipo de información que se desea tener a la salida dependiendo del tipo de estudio que se quiera realizar.

## 2.3 Estructura de la información

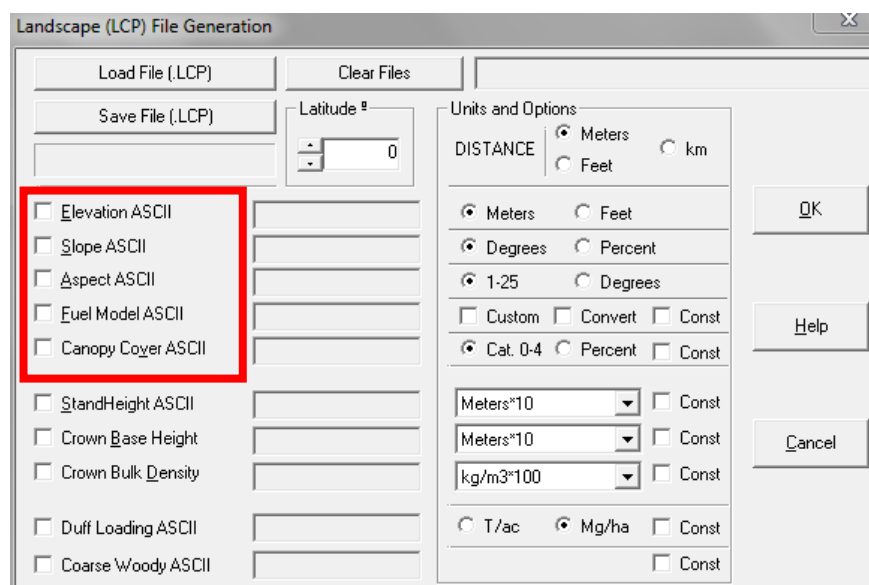
Como bien se ha comentado en el apartado anterior para que Farsite se pueda ejecutar requiere unos datos de entrada.

En este apartado se pasarán a describir las opciones de los ficheros y los parámetros de entrada que se han utilizado para realizar este proyecto. De la misma manera, también se describirán los ficheros de salida que se han escogido para poder analizar la simulación.

### 2.1.1 Ficheros de entrada

Farsite necesita cinco de los diez ficheros rasters que se le pueden introducir para llevar a cabo su simulación.

Dichas capas tienen que tener un formato ASCII para que de esta manera puedan ser reconocidas por el programa. Cada uno de estos ficheros rasters ha de tener la misma cabecera, donde en ella se obtiene la información de: el número de filas y columnas que forman el fichero, el tamaño de cada celda, la coordenada UTM, X e Y que identifica la posición de la zona a tratar. Estos cinco rasters se pueden observar en la siguiente figura (**Fig.2.3**):



**Fig. 2.3** Rasters de entrada necesarios para la simulación.

- **Altura del terreno (*Elevation*):** Representa la altura sobre el mar que tiene cada punto de la zona a tratar. Esta información es importante para el ajuste adiabático de temperatura y humedad y para la representación de la propagación del fuego entre las horizontales y las pendientes de la zona.
- **Pendiente (*Slope*):** Aporta la información de los grados que tiene cada pendiente de la zona a estudiar respecto a su horizontal. El tema pendiente es necesario para calcular los efectos que tiene la pendiente sobre la extensión del fuego y la radiación solar.
- **Aspecto (*Aspect*):** Aporta la orientación de las pendientes sobre el terreno. Para ello indica los grados que esta desplazada la pendiente siguiendo las agujas del reloj.
- **Fracción de superficie cubierta (*Canopy*):** Es el porcentaje que representa la cantidad de coronas de árboles que se encuentran en la horizontal de la superficie del suelo.
- **Combustible (*Fuel*):** Este raster debe especificar el número entero que representa el tipo de modelo de combustible. Este número entero tiene que ser un número entre 1-13 ya que se basan en el estándar original de la tabla de modelos de combustibles de NFFL (Anderson, 1983) (ver **Tabla 2.1**).

**Tabla 2.1.** Modelos de combustibles de NFFL (Anderson, 1983) [21].

Code	Description
1	Short grass (1 foot)
2	Timber (grass and understory)
3	Tall grass (2.5 feet)
4	Chapparral (6 feet)
5	Brush (2 feet)
6	Domant brush, hardwood slash
7	Southern rough
8	Closed timber litter
9	Hardwood litter
10	Timber (litter and understory)
11	Light logging slash
12	Medium logging slash
13	Heavy logging slash

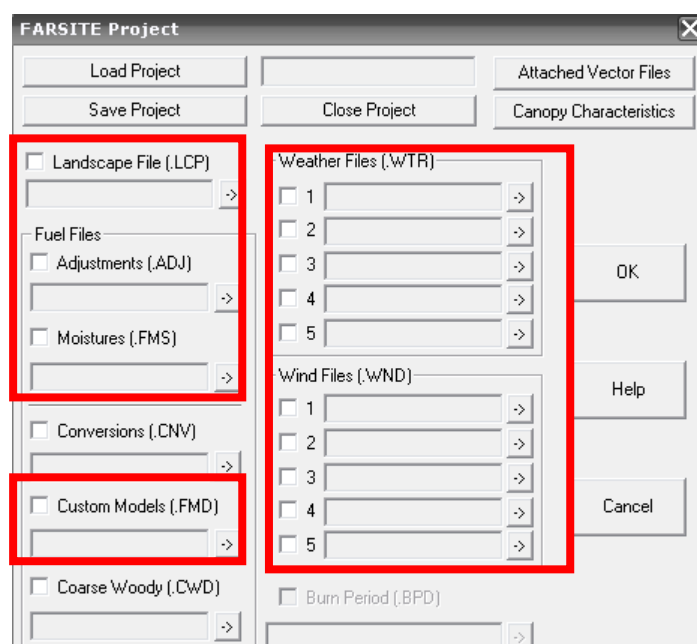
El significado de cada valor de la tabla anterior se encuentra en el anexo **Annexo 1**.

En la **Fig.2.4**, se puede ver visualmente el significado de cada uno de los rasters nombrados anteriormente.



**Fig.2.4** Representación de los rasters de entrada en un entorno real [22].

Además de los ficheros nombrados anteriormente, Farsite también necesita que se le introduzcan ficheros con información meteorológica, ya que es muy importante para la simulación saber qué tiempo hace. Hay que pensar que un incendio no actúa de la misma manera si en el momento del incendio viene viento del norte o del sur. Los ficheros meteorológicos necesarios para la simulación son los que se pueden observar en la siguiente imagen (**Fig.2.5**):



**Fig. 2.5** Ficheros meteorológicos de entrada.

- **Ajustes (*Adjustments*):** Ajusta la velocidad a la que el fuego se tiene que propagar dependiendo para cada modelo de combustible.
- **Humedades (*Moistures*):** Determina la humedad de combustible al comienzo de la simulación que se debe establecer para cada modelo de combustible. Dicha humedad es necesaria para iniciar el proceso de cálculo de humedad en un sitio específico de combustible en cada intervalo de tiempo durante la simulación.
- **Modelos personalizados (*Custom models*):** Especifica el tipo de comportamiento que tiene cada uno de los diferentes modelos de combustible que se definen en la **Tabla 2.1**. También hay la opción de crear nuevos modelos de combustibles especificando el comportamiento de ellos.
- **Archivos del tiempo (*Weather Files*):** Contiene observaciones diarias de la temperatura, humedad y precipitaciones en el periodo anterior y durante la simulación.
- **Archivos de viento (*Wind Files*):** Determina las características de viento que hará durante el periodo de la simulación. Los vientos son generalmente variables en el espacio y el tiempo. Farsite sin embargo, asume que los vientos son constantes en el espacio para una corriente de viento pero variable en el tiempo. Es decir, no hay efectos topográficos sobre los vientos.

Como se ha podido observar en la **Fig.2.5** los archivos del tiempo y del viento son los únicos que tienen la opción de introducir más de uno en la misma simulación, esto es debido a que como el tiempo cambia



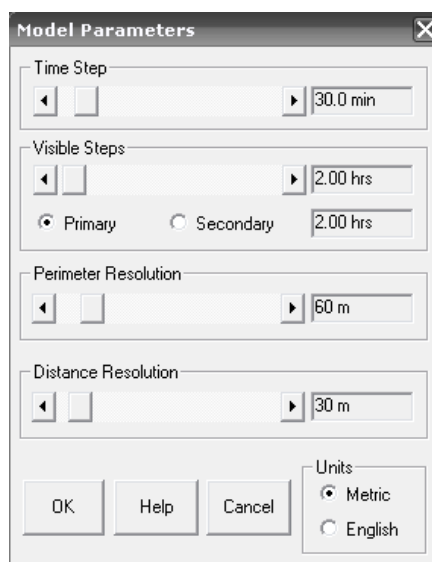
constantemente se tiene la posibilidad de introducir por separado el tiempo y los vientos de cada uno de los días que se quiera simular hasta un máximo de 5.

### 2.1.2 Parámetros de entrada

Para llevar a cabo la simulación además de los rasters de entrada que se han comentado anteriormente, hacen falta otro tipo de parámetros de entrada para así poder determinar por ejemplo el tiempo de la simulación u otro tipo de información.

- **Parámetros del modelo**

Con esta opción se especifica el avance que tendrá la simulación tanto de forma temporal como espacial. Los diferentes parámetros que componen ésta opción se pueden observar en la **Fig.2.6**.



**Fig. 2.6** Parámetros de modelo.

La definición de cada campo de “Parámetros de modelo” es la siguiente:

- Tiempo de paso (Time Step) : Cantidad de tiempo que las condiciones son constantes en un punto determinado para que la posición del frente del fuego se pueda proyectar.
- Paso de tiempo visible (Visible Steps) : El paso de tiempo visible es siempre un múltiplo del paso de tiempo real, no puede ser más corto que éste. El frente del fuego se dibuja durante la simulación en los intervalos de paso visible del tiempo indicados. El paso de tiempo es muchas

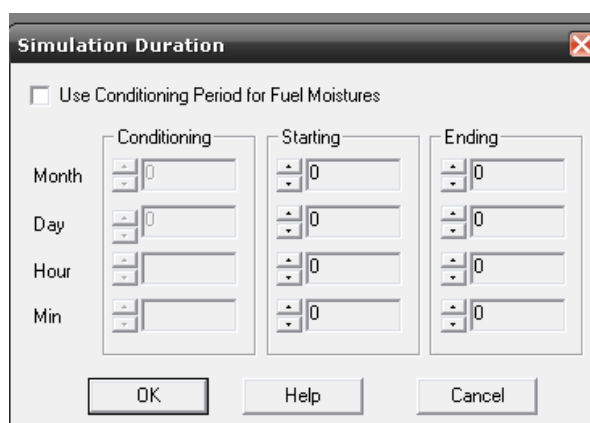
veces más largo que el paso de tiempo real para evitar detalles innecesarios en posiciones perímetro del incendio.

- Resolución del perímetro (*Perimeter Resolution*) : La Resolución del perímetro define el espacio máximo de puntos de proyección a lo largo del perímetro del incendio. Por ejemplo, una resolución del perímetro más pequeño genera más puntos de proyección del fuego.
- Distance Resolution : La resolución define la distancia máxima que el fuego puede viajar en un intervalo de tiempo sin tener nueva información del paisaje.

Las unidades que se han escogido para este proyecto son las métricas.

### • Duración de la simulación

Esta opción define el periodo de tiempo que se va a simular. Los parámetros que componen dicha opción se pueden observar en la **Fig.2.7**.



**Fig. 2.7** Duración de la simulación.

Se define el mes, día, hora y minuto que empieza y termina la simulación. Estos datos tienen que ser coherentes con los datos meteorológicos que se han comentado anteriormente. El programa necesita tener datos meteorológicos de unos días antes de que empiece la simulación hasta su final ya que así calcula la humedad y el tiempo del lugar.

En la **Fig.2.7** se puede observar como hay 3 columnas: acondicionamiento (conditioning), comienzo (starting), final (ending). Con el comienzo y el final se selecciona el mes, días, hora y minuto en el que se quiera empezar y terminar la simulación respectivamente. Con la opción acondicionamiento (opción no obligatoria), se selecciona el mes y el día des del cual se quiera recoger información del tiempo y las condiciones en las que se encuentre la tierra (humedad, temperatura, etc.).

### 2.1.3 Ficheros de salida

Como ya se comentó en el capítulo anterior, Farsite genera unos ficheros de salida durante su simulación que son los que aportan la información trascendente para poder estudiar el incendio. Farsite puede generar ocho tipos distintos de ficheros de salida, pero todos ellos se tienen que haber seleccionado previamente antes de iniciar la simulación.

Para llevar a cabo este proyecto solo se han hecho uso de dos de los ocho ficheros de salida: tiempo de llegada (TOA, *time of arrival*) y calor por unidad de área (HPA, *heat for area*) (ver Fig.2.8).

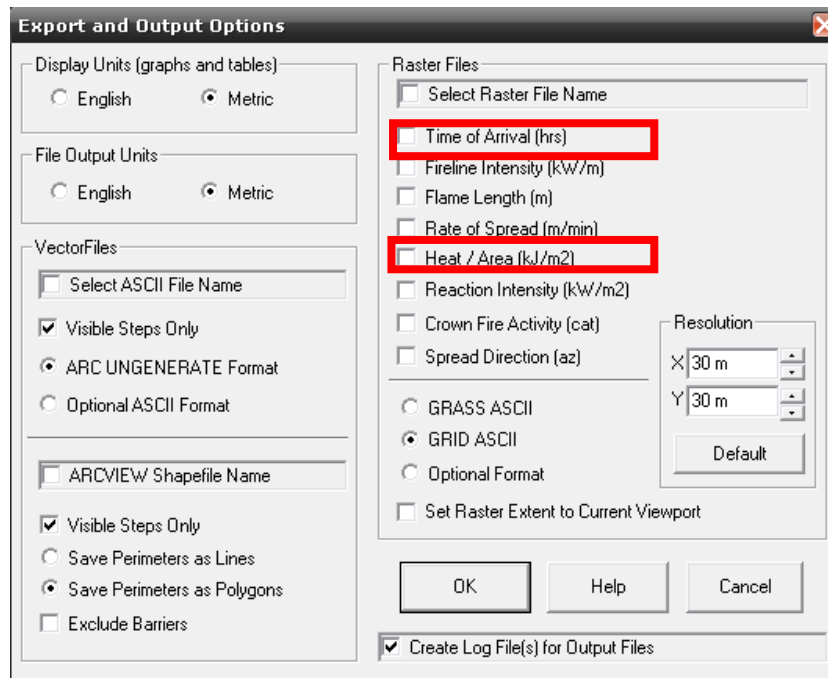


Fig. 2.8 Ficheros de salida.

- **Tiempo de llegada (TOA, *time of arrival*):** Aporta la información del tiempo (en unidades de horas) en el que ha llegado el incendio en un punto determinado.
- **Calor por unidad de área (HPA, *head for area*):** Contiene información de la calor en un punto determinado. Las unidades que utiliza son KJ/m2.

Dichos ficheros de salida se generan en formato raster.

## CAPÍTULO 3. GENERACIÓN DE LOS RASTERS PARA UN INCENDIO EN CATALUÑA NORTE

### 3.1 Creación de los rasters de entrada para el simulador Farsite

Como se ha comentado en el capítulo anterior, Farsite necesita cinco capas rasters para que se pueda iniciar.

Los profesionales que utilizan Farsite como herramienta de trabajo y están más familiarizados con él, utilizan el programa Arcview [23] para generar los diferentes rasters a través de un mismo mapa. Pero ya que dicho programa es muy costoso y en este proyecto solo se van a realizar pruebas, se ha decidido crear los rasters de otra manera.

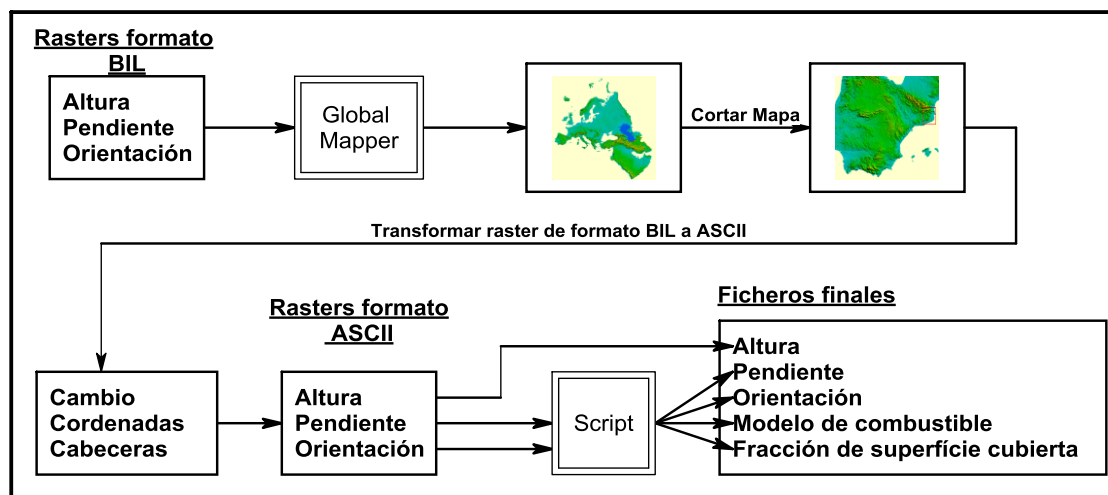
Los rasters correspondientes al aspecto, elevación y pendiente del terreno, se han conseguido mediante la organización USGS (U.S Geological Survey) [24], donde dichos rasters se encuentran clasificados en continentes. Para este proyecto se han utilizado los tres rasters nombrados anteriormente correspondientes al continente de Europa. Dichos rasters se encuentran en formato .BIL.

Debido a que Farsite necesita que los rasters se introduzcan en formato ASCII, se ha hecho uso del programa Global Mapper [25] tanto para hacer la transformación de formato BIL a ASCII, como para cortarlos y obtener solo la zona deseada, que en nuestro caso ha sido la comarca de Girona y el norte de Barcelona.

Una vez ya se han transformado los ficheros en formato ASCII se tienen que modificar los valores de los ficheros slope y aspect, ya que los valores de dichos rasters facilitados por USGS están aumentados en un factor de 100. Por lo tanto se ha realizado un script que accede a los ficheros slope y aspect y divide cada uno de los valores de dichos ficheros entre 100.

Por último, para poder crear los otros dos rasters que se necesitan (fuel y canopy), se ha ampliado el script comentado anteriormente, para que solo ejecutándolo se generen. Esta parte del script lo que hace es asignar a cada píxel de los dos rasters que se tienen que generar, el valor que le corresponda según la información facilitada por el CREAF (Centre de Recerca Ecològica i Aplicacions Forestals) [26].

En la **Fig.3.1** se puede observar todo el procedimiento que siguen los rasters hasta que se consigue obtenerlos de forma deseada para introducirlos a Farsite.



**Fig. 3.1** Procedimiento de transformación de los rasters de entrada.

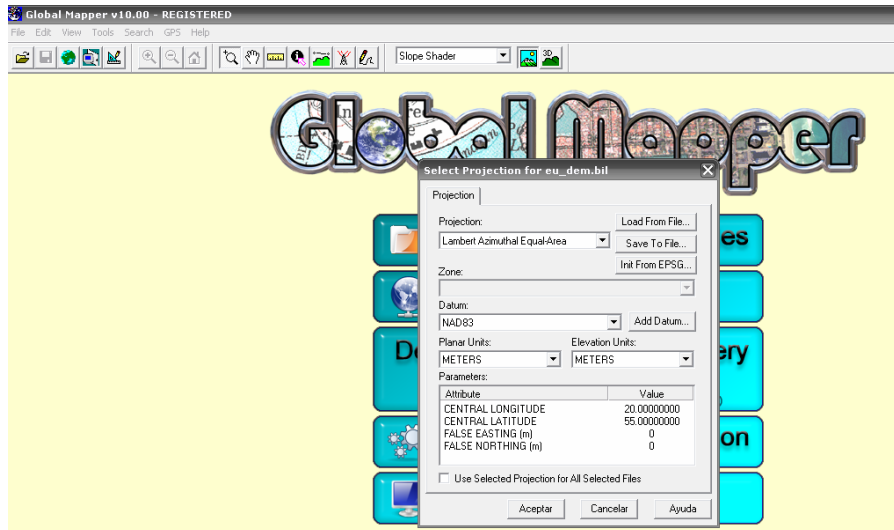
En los apartados siguientes se va a explicar la realización de cada uno de los rasters de forma más detallada.

### 3.1.1 Global Mapper

El programa Global Mapper es un programa que permite visualizar todo tipo de mapas en diferentes formatos. Tiene muchas funcionalidades, pero en este proyecto se ha utilizado básicamente para las siguientes: cambiar el formato del mapa, cortarlo por la zona deseada y hacer el cambio de coordenadas Lambert azimuthal equal-area a grados.

Todas estas cosas y más son las que se van a explicar en este apartado.

Para que el programa Global Mapper permitiera abrir el raster en formato .BIL requería la introducción de diferentes parámetros (Proyeccion, Datum, Unidades utilizadas, longitud y latitud central, etc.), tal y como se puede observar en la **Fig.3.2**:

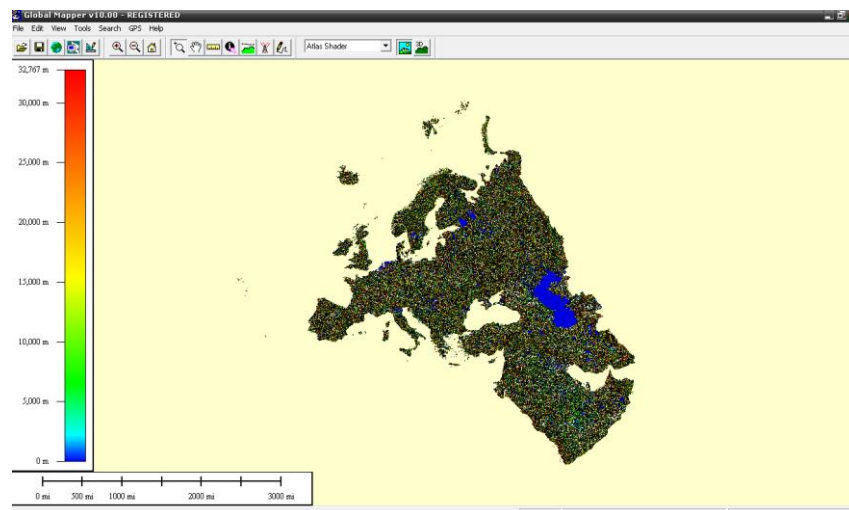


**Fig. 3.2** Parámetros necesarios que el programa Global Mapper necesita para poder abrir los ficheros.

El valor de los parámetros que se han introducido los ha facilitado la página el proveedor de los datos.

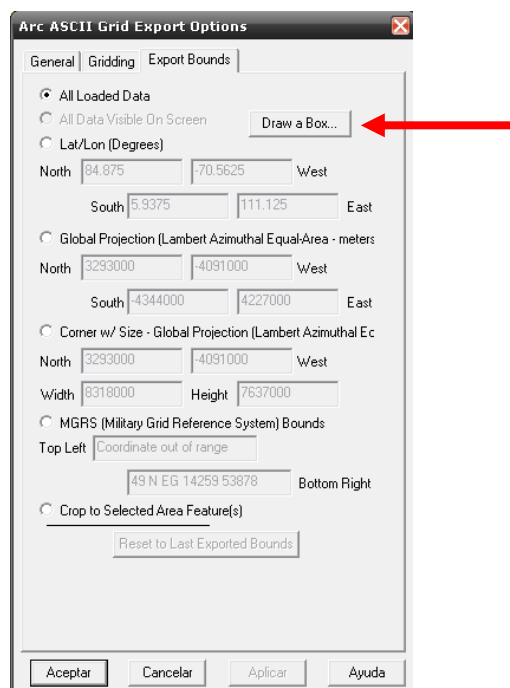
En este caso el tipo de proyección que se utiliza es la *"Lambert Azimuthal Equal-Area"*. El punto central de todo el mapa de Europa está situado a : 20° 00' 00"E de longitud Central y 55° 00' 00"N de latitud central.

Una vez se han introducido los valores necesarios, ya se puede visualizar el mapa en el programa (ver **Fig 3.3**).



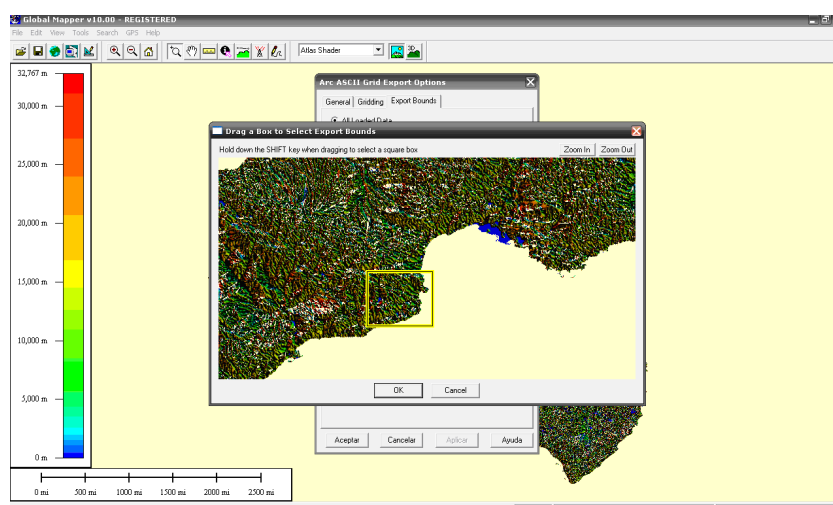
**Fig. 3.3** Visualización del fichero de Europa en formato .BIL con el programa Global Mapper .

Seguidamente ya se puede pasar a transformar el fichero a formato ASCII, y a su vez cortarlo por la zona deseada. Cuando se desea exportar el fichero de formato BIL a ASCII, aparece una opción para poder cortar el mapa, tal y como se puede observar en la **Fig.3.4**.



**Fig. 3.4** Opción para cortar y exportar a otro formato el fichero en formato BIL.

Si se selecciona la opción de “*Draw a Box..*” se podrá especificar de forma grafica la parte del mapa que se desea cortar (ver **Fig.3.5**)



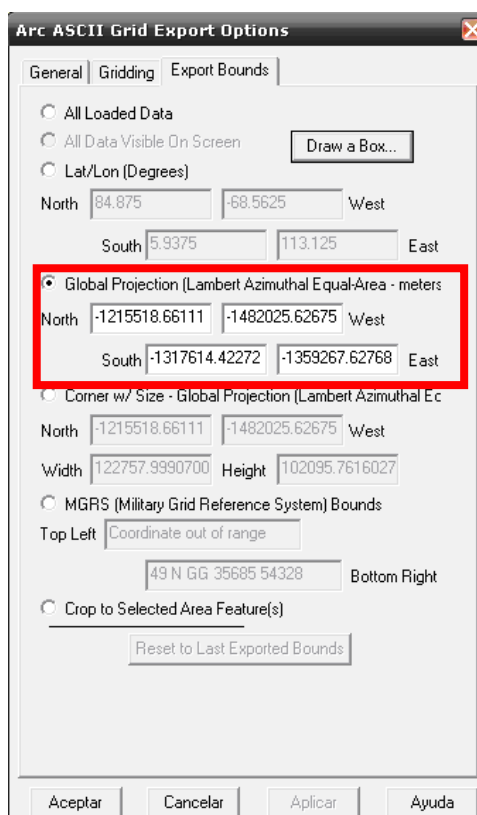
**Fig. 3.5** Zona del mapa a cortar.

Si observamos detalladamente la parte del mapa a cortar, se puede identificar que la zona seleccionada, corresponde al Pirineo Oriental, en gran parte a la provincia de Gerona (ver **Fig.3.6**).



**Fig. 3.6** Representación de la zona del mapa cortada.

Aunque se haya realizado el corte de forma manual, se puede observar como en las opciones anteriormente vistas, ahora aparecen los puntos exactos por donde se ha cortado el mapa (ver **Fig.3.7**).

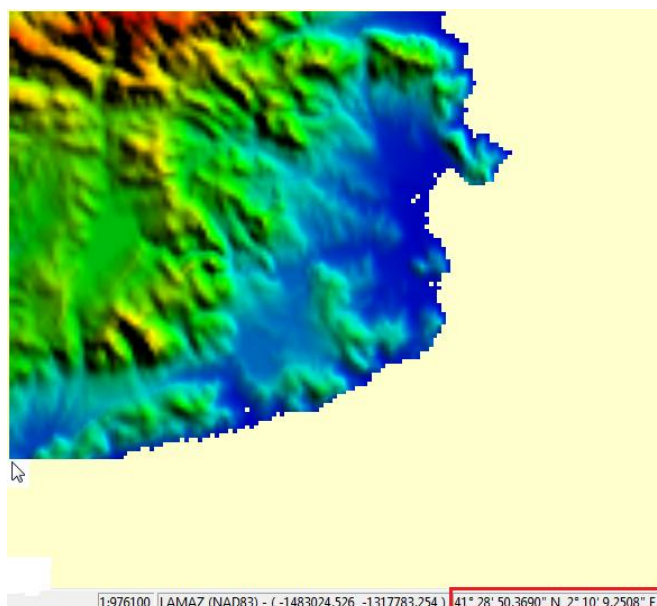


**Fig. 3.7** Puntos de corte de la zona seleccionada en el mapa.



Si se toma nota de dichos puntos, se pueden introducir en la opción “*Global Projection*” de los otros dos rasters que aún quedan para cortar. De esta manera los tres rasters tendrán exactamente las mismas dimensiones, ya que tendrán los mismos puntos de corte. Una vez ya se han cortado los tres rasters, se tiene que abrir uno con el programa global Mapper introduciendo los mismos parámetros que se han visto en la **Fig.3.2**.

Después de visualizar el raster en el programa, se tiene que cambiar el formato de coordenadas que se utiliza el fichero, esto es preciso realizarlo debido a que después el simulador Farsite trabaja de forma más precisa y correcta con coordenadas UTM que con las coordenadas Lambert Azimuthal Equal-Area o las GPS. Para realizar dicho cambio se tiene que abrir de nuevo el fichero con el programa Global Mapper y una vez abierto colocar el puntero del ordenador sobre el marco de abajo a la izquierda del raster, de esta manera se podrá observar como en la barra de abajo del programa aparecen las coordenadas de dicho punto en formato GPS, tal y como se puede observar en la **Fig.3.8**.



**Fig. 3.8** Obtención de la coordenada GPS.

Después de obtener dicha coordenada en formato GPS se tiene que transformar a formato UTM mediante la herramienta de conversión que ofrece el “Institut Cartogràfic de Catalunya” a través de internet [27] .

Una vez obtenidas dichas coordenadas en formato UTM se tienen que abrir cada uno de los tres rasters cortados y modificar los campos de la cabecera pertenecientes a xllcorner y yllcorner, tal y como se puede observar en la **Fig.3.9**.

ncols	124	➔	ncols	124
nrows	103		nrows	103
xllcorner	-1482525.627		xllcorner	430762.39
yllcorner	-1318018.661		yllcorner	4592270.16
cellsize	1000		cellsize	1000
nodata_value	-9999	nodata_value	-9999	

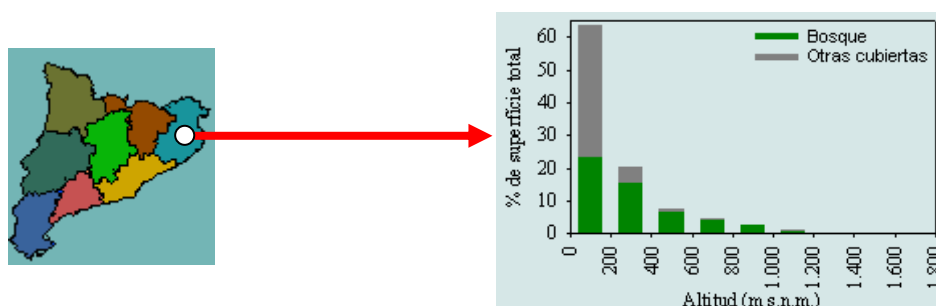
**Fig. 3.9** Cambio del campo xllcorner y yllcorner de la cabecera de los rasters.

### 3.1.2 Realización del Script para la creación de los rasters *canopy* y *fuel*

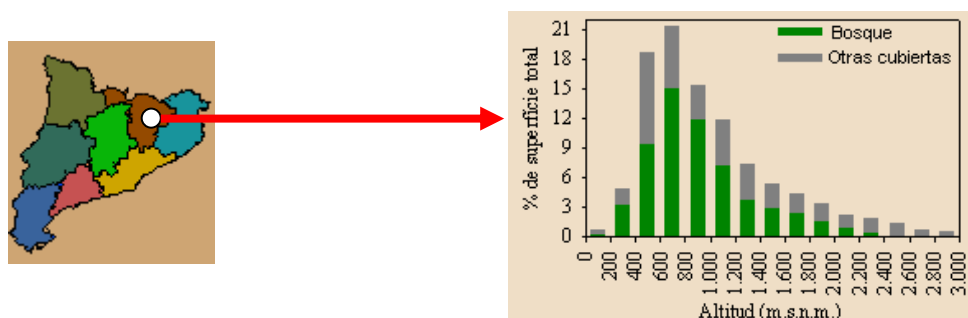
Para realizar el script y que la información que aporten los rasters de *canopy* y *fuel* sea coherente y válida, se ha utilizado información proporcionada por el CREAM, como bien ya se ha comentado anteriormente.

- **Raster *canopy***

Para generar el raster de *canopy* se han utilizado dos graficas (**Fig.3.10** y **Fig.3.11**), donde en ellas se representa el porcentaje de vegetación que hay en la zona seleccionada para realizar el proyecto (provincia de Girona), en función a la elevación del terreno.



**Fig. 3.10** Distribución de las cubiertas de la tierra según la altitud del terreno correspondiente a las regiones: *Alt Emporda, Gironès, Baix Emporda, Pla de l'Estany y la Selva*.

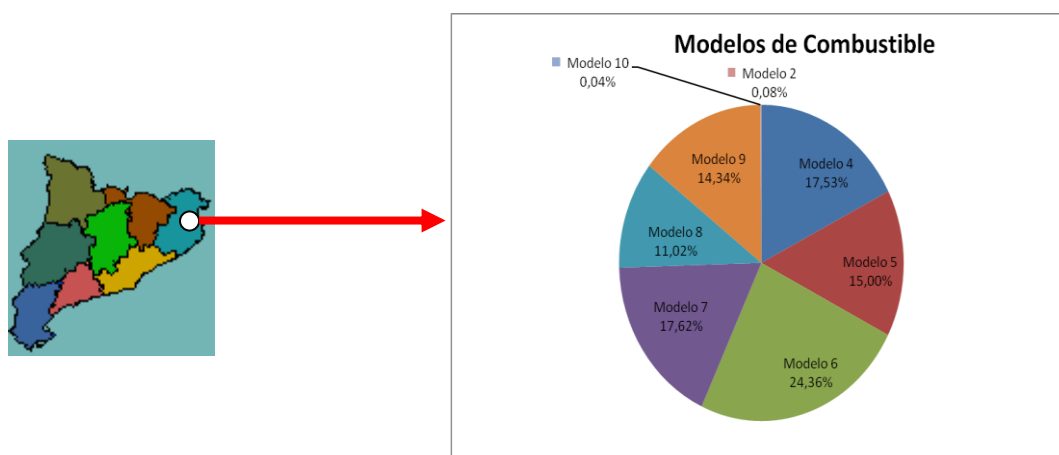


**Fig. 3.11** Distribución de las cubiertas de la tierra según la altitud del terreno correspondiente a las regiones: *Cerdanya, Garrotxa, Osona y Ripolles*.

Teniendo esta información lo que se ha hecho es generar el raster *canopy* a partir de los datos del raster *dem*. De esta manera, a cada píxel del raster *canopy* se le asigna un porcentaje de vegetación correspondiente al valor de dicho píxel en el raster *dem*. Es decir, se asigna un porcentaje de vegetación dependiendo de la altura que tenga cada punto. Para ello, como hay dos graficas con valores distintos para la misma elevación, se ha hecho una media ponderada de los porcentajes de ambas.

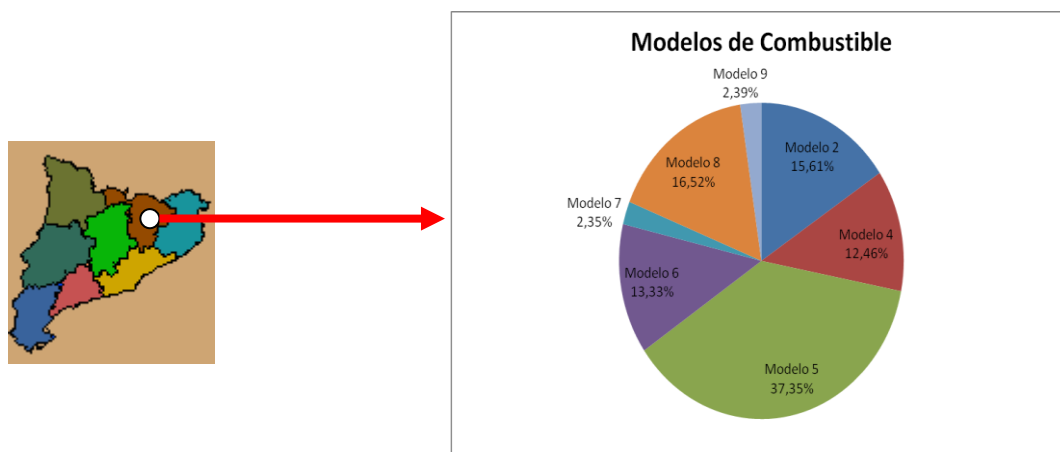
- **Raster *fuel***

Para generar el raster *fuel* se ha utilizado la información que aportan las gráficas que se pueden ver a continuación (**Fig.3.12** y **Fig.3.13**), los cuales indican el porcentaje del tipo de combustible que se encuentra en la zona seleccionada. Estas graficas se han obtenido de la misma manera que las del raster *canopy*, es decir a través del CREAM.



El 74,5% de los modelos representados en el gráfico anterior son el grupo de los matorrales (modelos 4,5,6 y 7), siendo mayoritarios los modelos 6, 4 y 7, la cual cosa indica la elevada carga de combustible en gran parte de los bosques de esta zona. El 25,4% restante corresponde al grupo de la hojarasca (modelos 8,9 y 10).

**Fig. 3.12** Distribución de los modelos de combustible correspondiente a las siguientes regiones: *Alt Emporda*, *Gironès*, *Baix Emporda*, *Pla de l'Estany* y *la Selva*.



El 65% de la superficie arboleada de la región tiene modelos de matorral (4, 5, 6 y 7) de los cuales, el modelo 5 es claramente mayoritario, el 37% del total. Un 35% son modelos de pastos (15%) o de hojarasca (19%), sobretudo el modelo 8.

**Fig. 3.13** Distribución de los modelos de combustible correspondiente a las siguientes regiones: *Cerdanya, Garrotxa, Osona y Ripolles*.

Para ello se ha hecho una media de los porcentajes de las dos graficas, y una vez obtenidos los resultados se han tenido en cuenta los porcentajes del raster *canopy* y se han ido asignando los modelos con mayor rango de porcentaje, a los píxeles donde en el *canopy* también tienen el mayor porcentaje.

Se ha unido en un solo código, el script que genera los rasters *canopy* y *fuel*, y el que modifica el contenido de los rasters *slope* y *aspect*. Dicho Script se puede encontrar en el **Annexo 2**.

En el **Annexo 3** se puede ver el resultado de dicho Script, es decir, los diferentes rasters que se han utilizado en este proyecto para poder entrar al simulador Farsite.

## 3.2 Pruebas

Una vez ya se tienen claros los conceptos del simulador Farsite y se han generado los ficheros necesarios de forma adecuada, se ha pasado a realizar un conjunto de pruebas para comprobar su correcto funcionamiento.

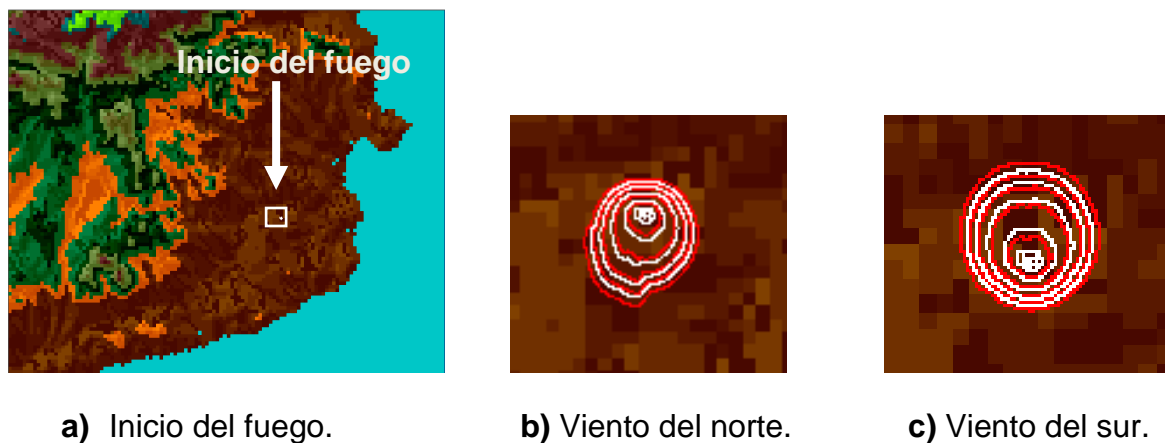
### 3.2.1 Prueba 1: Efectos del viento

Para poder observar como realmente influyen los ficheros meteorológicos en el comportamiento de un incendio, en esta primera prueba se han generado dos incendios idénticos, dónde la única diferencia entre ellos es la dirección de la

que proviene el viento. Como se puede comprobar en la **Fig 3.14** en el apartado “**a**)” se muestra el inicio del fuego de las dos simulaciones realizadas.

Para poder modificar de donde proviene el viento, el fichero que se ha tenido que modificar es el fichero meteorológico *Wind File*. En él se indica de donde proviene el viento mediante los grados de una circunferencia, es decir, si el viento proviene del norte el valor de dicho campo valdrá 0 y si el viento proviene del sur el valor valdrá 180 (media circunferencia).

En la **Fig.3.14** apartado “**b**)” se puede observar como gran parte del incendio se ha extendido hacia el sur, esto es debido a que el viento en esta simulación provenía del norte. Por otra parte, si observamos el apartado “**c**)” de dicha figura tiene cierto parecido a la extensión del incendio del apartado “**b**)” pero en este caso a la inversa, es decir, el fuego se ha extendido hacia el norte, hecho provocado por el viento, que en este caso provenía del sur



**Fig. 3.14** Efectos del viento.

Si se comparan las dos simulaciones de forma exacta se puede ver como el incendio de la simulación del apartado “**c**)” tiene una forma más redondeada que el del apartado “**b**)”, esto es debido a que en el del apartado “**b**)” hay algún tipo de desnivel en el terreno que hace que se propague de forma ligeramente diferente, este hecho se explicara más adelante en otra prueba.

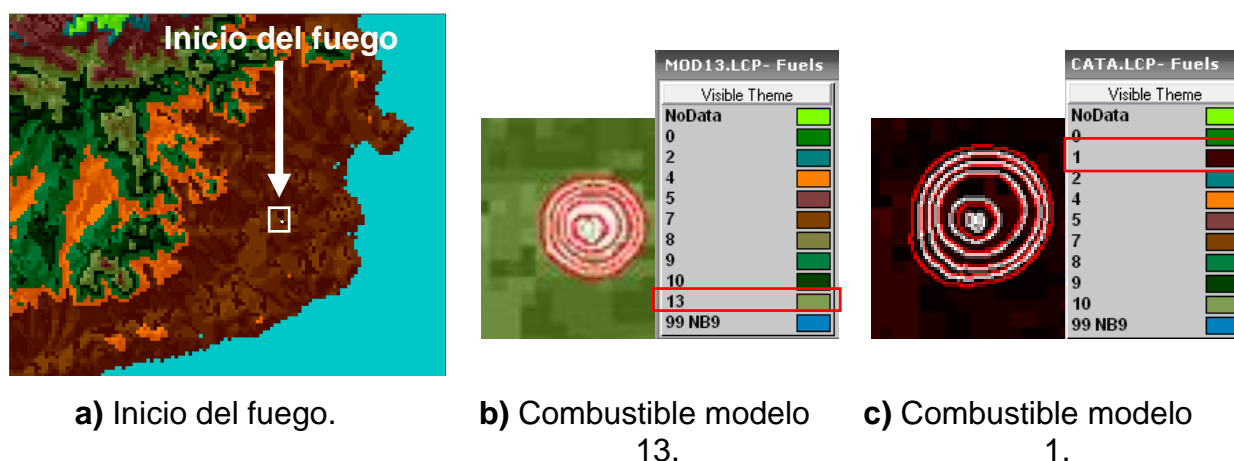
### 3.2.2 Prueba 2: Efecto del tipo de combustible

En esta segunda prueba se ha querido observar el comportamiento del fuego con solo cambiar el tipo del modelo de combustible a quemar.

Para ello se han realizado dos simulaciones idénticas exceptuando que en una se ha quemado una zona de modelo de combustible 13 que significa que en el terreno hay pasto fino, seco y bajo, y en la otra simulación se quema exactamente la misma zona cambiando en este segundo caso el modelo

combustible 13 por el modelo 1, donde el modelo 1 indica que en el terreno hay acumulaciones de restos gruesos y pesados. Esta información se puede encontrar en **Tabla 2.1** y/o en el **Anexo 1**.

Para poder cambiar los tipos de combustible se ha tenido que modificar el raster fuel.



**Fig. 3.15** Efectos del tipo de combustible.

En la **Fig 3.15 “a)”** se puede observar donde se ha iniciado el incendio en las dos simulaciones.

Como se puede comprobar en el apartado “b)” de dicha figura el incendio no se ha extendido demasiado debido a que se ha quemado una zona con modelo de combustible 13, el cual es un modelo que no se quema fácilmente.

Por otra parte como se puede observar en el apartado “c)” el incendio se ha extendido mucho más comparado con el del apartado “b)”, esto es debido a que en este caso se ha quemado un modelo de combustible del tipo 1 el cual hace que el fuego se propague de forma rápida.

### 3.2.3 Prueba 3: Efecto de la orografía

En esta última prueba se ha querido comprobar la diferencia de comportamiento que tiene un incendio dependiendo del tipo de zona que se queme.

Para ello, se han iniciado dos inicios de fuego en la misma simulación, tal y como se puede observar en la **Fig.3.16**. Un incendio se inicia en la parte de montaña (apartado “b)” ), donde se encuentran diferentes desniveles, y el otro se inicia cerca de la costa (apartado “c)” ) donde no hay ningún tipo de desnivel.



**Fig. 3.16** Efectos de la orografía.

Como se puede comprobar en los apartados “b)” y “c)” de la **Fig.3.16** los dos incendios han evolucionado de forma distinta entre ellos. El incendio originado en la montaña se ha extendido de forma desigual, ya que debido a los diferentes tipos de elevación que tiene a su alrededor le ha limitado su velocidad de extensión, de la misma manera que los diferentes tipo de combustible que le rodean, que como bien se ha podido comprobar en el apartado anterior también pueden provocar que el incendio se extienda a diferentes velocidades. En cambio si se observa el incendio más cercano a la costa se ha extendido uniformemente ya que no tenía ningún tipo de desnivel y el modelo de combustible es en todo su proceso del mismo tipo, debido a ello se ha ido extendiendo con la misma rapidez por todo su alrededor.

Realizando estas pruebas se ha querido comprobar la importancia que tienen los efectos meteorológicos, el tipo de vegetación y la altura de la zona en el comportamiento de cualquier incendio.

## CAPÍTULO 4. EL USO DE UAS EN INCENDIOS

### 4.1 Servicios diseñados para los puntos calientes

Cuando se origina un incendio forestal, los especialistas en extinción de incendios lo primero que intentan es controlar el fuego de manera que este no se pueda extender más de lo que ya se ha extendido. Una vez esta controlado y prácticamente apagado son muchos los puntos calientes del terreno que continúan teniendo una temperatura elevada y por lo tanto la zona del incendio tiene que estar vigilada durante un cierto tiempo para ir apagando dichos puntos y así evitar que se vuelva a reactivar.

El problema está en que una vez esta apagado el incendio es difícil definir los puntos que quedan calientes en la zona y más si la zona quemada es considerablemente extensa. Son muchas las personas que se necesitan para poder localizar los puntos calientes y el equipo de tierra que se utiliza es muy costoso y lento.

Este proyecto quiere ser una solución al problema nombrado anteriormente. Es mucho más sencillo y eficazmente rápido que un avión dotado de una cámara térmica sobrevuele el perímetro del incendio e informe a la estación de tierra de la coordenada exacta en la que hay un punto caliente, de esta manera el equipo de extinción podrá ir rápidamente al punto exacto para apagarlo.

Además una de las ventajas que tiene este sistema es que aun que sea de noche puede realizar el mismo servicio y con la misma eficacia que si fuera de día, hecho que actualmente no se puede realizar, o no con la misma eficiencia.

El equipo de investigación ICARUS de la Escuela Politécnica Superior de Castelldefels trabaja sobre una arquitectura distribuida tanto en el UAS como en la estación de tierra, y utilizan el sistema Marea para que los diferentes servicios que forman esta arquitectura se puedan comunicar.

### 4.2 Marea

MAREA (Middleware Architecture for Remote Embedded Applications) es un middleware desarrollado por el equipo de investigación ICARUS. Un middleware es un software de comunicación que se utiliza para ofrecer diferentes servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas.

Se desarrolla como una capa de abstracción de software, la cual se sitúa entre la capa de aplicaciones y la capa de red [28], de esta manera permite conectar entre sí a diferentes productos de diferentes proveedores.



Existen diferentes tipos de middleware, pero el que se ha utilizado en este proyecto es del tipo CORBA (Common Object Request Broker Arquitectura) y se basa en una red de cooperación de los diferentes servicios conectados a ella, en este caso, de las diferentes aplicaciones. Estos servicios envían y reciben datos hacia y para otros servicios conectados al mismo, de esta manera entre ellos se pueden ir complementando [29].

La ubicación de los servicios no es importante, ya que el middleware gestiona su descubrimiento. De la misma manera, también se ocupa de todas las tareas de transferencia: direccionamiento de mensajes de datos, cálculos de ruta, entrega, control de flujo, reintentos, etc. En definitiva, se ocupa de todo lo necesario para que los servicios conectados se puedan comunicar entre sí sin ningún tipo de problema. Tiene una arquitectura Publish/Suscribe consiste en el envío de mensajes entre un elemento que actúa como “publicador” de la información y un elemento que actúa como “suscriptor” de la misma. Cualquier servicio puede ser publicador, suscriptor o ambos a la vez.

Gracias al Sistema de Distribución de Datos (DDS) este middleware puede ofrecer el modelo de publicación-suscripción que elimina virtualmente la complejidad de programación de la red para aplicaciones distribuidas. Para ello la capa del middleware es la que se ocupa de recibir la información que envían los nodos que se han publicado, hacia los nodos que se han suscrito a dicha información. Esto es posible, gracias a que el middleware ofrece 4 primitivas de servicio, para poder realizar las diferentes comunicaciones: variables, eventos, funciones y ficheros.

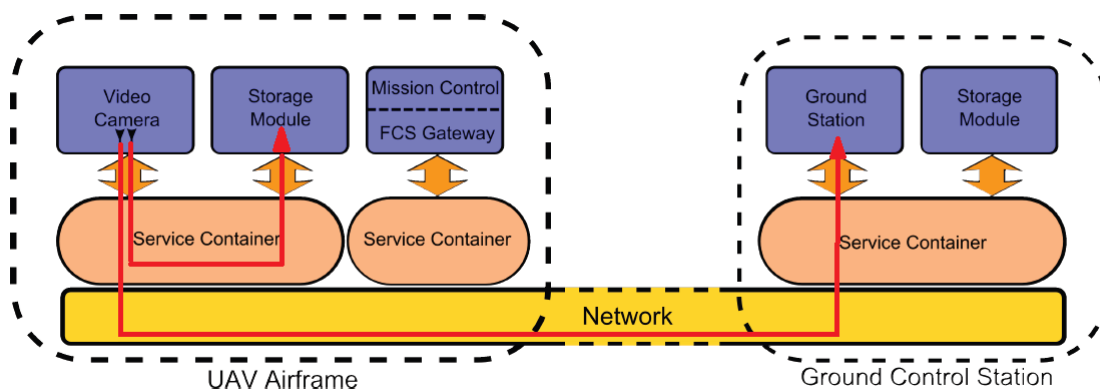
- Variables: Envían información periódica utilizando el protocolo UDP ya que en este caso es más importante que se envíe y se reciba la información de forma rápida que no que se pueda perder un paquete. En este proyecto se ha utilizado para recibir las coordenadas GPS del UAV.
- Eventos: En este caso se utiliza el protocolo TCP para enviar y recibir la información ya que esta primitiva envía información importante, como por ejemplo podría enviar una alarma en el caso de que el UAV se estrelle.
- Funciones: La comunicación que se utiliza en este caso es bidireccional y para ello utiliza el protocolo TCP, ya que se realiza una petición de la información que le será entregada por el servicio requerido y por tanto la información no se puede perder. Un ejemplo de su utilización sería el envío del Plan de vuelo del avión.
- Ficheros: En este caso la primitiva se utiliza para enviar ficheros de información de gran tamaño. Para ello utiliza el protocolo UDP ya que es el más rápido, pero como en este caso no interesa que se pierda ningún paquete los paquetes son numerados y si se pierde alguno se vuelve a reenviar. Por ejemplo se podría utilizar para el envío de imágenes de gran tamaño.

MAREA toma la forma de un componente llamado “*Service Container*”. Los servicios son ejecutados y gestionados por un contenedor de servicios que es único en cada nodo de la red del sistema distribuido.

Dicho contenedor puede gestionar varios servicios y proporciona funcionalidades comunes (acceso a la red, la entrega local de mensajes, la resolución de nombres y de almacenamiento en caché, etc.) de los servicios que contiene. Por tal motivo no es necesario que se conozca la ubicación de cada uno de los servicios que se ofrecen, si no que el contenedor se ocupa de saber quien entrega y recibe los mensajes, donde está cada servicio, y que sucede cuando un mensaje no ha sido recibido.

Como se puede observar en la **Fig. 4.1**, un servicio puede tener comunicación con los diferentes servicios conectados en el mismo contenedor de servicio que él, o bien con los demás servicios que están conectados a otros contenedores de servicios, siempre y cuando estos contenedores formen parte de la misma red que el contenedor de servicios al que está conectado.

Como por ejemplo se podría dar un caso en que el servicio de videocámara y el servicio de almacenamiento conectados al mismo contenedor tuvieran comunicación entre ellos y de la misma manera el servicio de videocámara puede estar conectado a su vez al servicio de la estación de tierra situado en otro contenedor distinto al suyo, gracias a estar conectados a la misma red. Este ejemplo se puede observar en la **Fig.4.1**.



**Fig. 4.1** Arquitectura del middleware.

Las funcionalidades que aporta un contenedor son:

- **Gestión de servicios:** El contenedor es el responsable de iniciar y detener los diferentes servicios que contiene. También se encarga de alertar a los otros servicios los cambios sobre el estado de un servicio mediante notificaciones.

- **Gestión de nombres:** En este caso la funcionalidad es muy parecida a la de un Proxy. Cada servicio tiene un nombre asignado, y el contenedor de nombres descubre la ubicación en la red del servicio mediante el nombre del servicio.
- **Gestión y abstracción de la red:** El middleware se encarga de gestionar las conexiones de los servicios a las diferentes redes. Controla las conexiones UDP/TCP y los grupos de broadcast.
- **Gestión de recursos:** Debido a que cada nodo de la red distribuida tiene un mismo contenedor y que todos los servicios de esa red están situados en la capa superior a la del contenedor, el contenedor es idóneo para centralizar la gestión de recursos compartidos: memoria, tiempo de CPU, dispositivos de entrada/salida que tienen acceso de forma exclusiva, etc.

### 4.3 Icarus Simulation Integrated Scenario

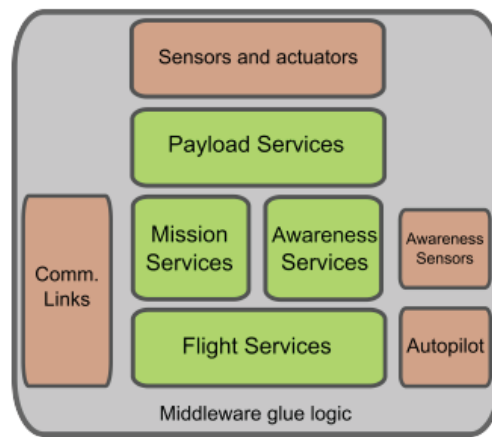
Icarus Simulation Integrated Scenario (ISIS) es una plataforma de simulación desarrollada para facilitar la realización de estudios y/o pruebas de UAS (*Unmanned Aircraft System*) mediante su simulación, ya que si se realizaran de forma real el coste sería muy elevado [30].

El objetivo de ISIS es presentar una plataforma de investigación que sea capaz de hacer frente a una variedad de misiones UAS con poco tiempo de configuración, reconstrucción y gastos generales. Además, esta plataforma no solo tiene que ser capaz de simular el comportamiento de los UAS desde el punto de vista de la misión, sino que además tiene que permitir incluir en ella otros vehículos adicionales.

Para ello ISIS implementa una arquitectura de simulación distribuida donde los vehículos y el entorno se simulan a través de un software especial. El middleware MAREA, como ya se ha comentado en el apartado anterior, es el que facilita la coordinación entre los diferentes vehículos y otras aplicaciones del sistema.

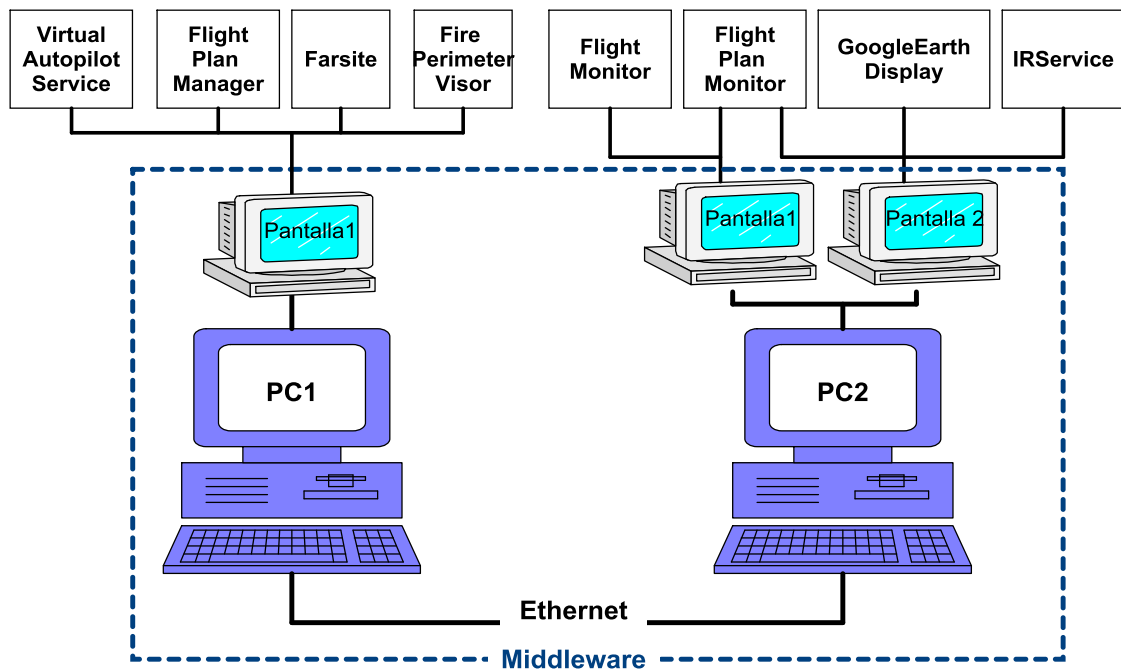
En la parte superior de este middleware se encuentra una capa de abstracción donde se han definido un conjunto de servicios reutilizables que cuentan con los elementos mínimos que se necesitan para cualquier misión de UAV (ver **Fig.4.2**).

Los servicios mínimos de los que se disponen para realizar las misiones se pueden clasificar en 4 categorías: Servicios de vuelo (*Flight Services*), Servicios de misión (*Misión services*), Servicios de carga útil (*Payload services*) y Servicios de alerta (*Awareness services*).



**Fig. 4.2** Servicios del UAS

Los servicios y la estructura de la plataforma ISIS que se ha utilizado para realizar este proyecto se pueden observar en la **Fig.4.3**.



**Fig. 4.3** Servicios y estructura de ISIS en el proyecto.

Como se ha podido observar en la **Fig.4.3** para realizar las simulaciones pertinentes para poder comprobar el funcionamiento del proyecto se han utilizado dos ordenadores donde en cada uno de ellos se utilizan distintos servicios. Gracias a la utilización de Marea y del simulador ISIS se pueden comunicar entre si y aparentar que se encuentran todos ellos en un mismo UAV.

El PC1 consta de cuatro servicios:

- **Virtual Autopilot Service:** Interactúa con el servicio Flight Monitor y cuando éste le da la orden se ejecuta el simulador Flight Gear.
- **Flight Plan Manager:** Este servicio es el responsable de recibir las coordenadas del plan de vuelo del UAV para transformarlas a waitpoints para que el Flight Gear pueda saber el camino a seguir.
- **Farsite:** Ejecuta un “bookmak” el cual solo abriéndolo permite ejecutar la simulación sin tener que introducir ningún dato de entrada debido a que en él se encuentran fijados todos los datos de entrada necesarios. Solo es necesario especificar los ficheros de salida.
- **Fire Perimeter Visor:** Permite crear un fichero del tipo KLM donde se encuentra definido el perímetro del incendio.

El PC2 está formado por dos pantallas y consta de cuatro servicios.

- **Flight Monitor:** Simula control de tierra del UAV y permite de alguna manera manejarlo pasando por sus diferentes estados ( stop, parking, take off..).
- **Flight Plan Monitor :** Permite cargar el plan de vuelo que simulara el Flight Gear.
- **GoogleEarth Display:** Ejecuta la herramienta Google Earth [31] en la que se puede ver el recorrido que va realizando el UAV.
- **IRService:** Es el responsable de recibir las coordenadas por las que va pasando el UAV e informar de la temperatura que hay en dicha coordenada.

Con ISIS se consiguen varios objetivos:

- Se define un hardware/software flexible y reutilizable que permite la abstracción de las diferentes funciones del UAS para desarrollar fácilmente las misiones. Esta arquitectura reutilizable permite cambiar los componentes de ISIS dependiendo de la misión que se está simulando. Se pueden añadir nuevos servicios que simulan el entorno, vehículos o sistemas, o bien volver a utilizar los mismos servicios que se hayan utilizado en misiones anteriores.
- Proporcionar una manera fácil y segura de realizar pruebas con los UAS. El hecho de no necesitar un UAS real ni los diferentes equipos necesarios para realizar las misiones hace que el coste de las pruebas se minimice y permita de esta manera realizar tantas como se necesite. Además, todos los problemas logísticos se evitan, y no hay riesgos adicionales causados por un mal comportamiento de cualquier elemento en la plataforma. Por último, todos los escenarios complejos pueden ser

probados sin importar el número de UAS, las condiciones meteorológicas, o parámetros UAS. Con esta arquitectura las maniobras peligrosas se pueden estudiar, hasta el límite de situaciones extremas como las colisiones.

- ISIS permite simular una misión de colaboración, es decir, varios vehículos en movimiento en torno al desarrollo de alguna acción en particular. El escenario sería prácticamente creado para este propósito. De esta manera no hay que gastar grandes cantidades de dinero para poner a prueba este caso en particular.
- Si se desea volver a repetir una simulación en concreto, ISIS puede volver a crear ese vuelo en particular para realizar un estudio más exhaustivo de él. Este nivel de simulación consiste en servir los mismos paquetes de telemetría que se tomaron anteriormente de la red. Los demás servicios no se darán cuenta si hay un vuelo real o simulado.

La idea de ISIS es minimizar el desarrollo de las pruebas y el coste de validación y proporcionar una fácil migración del software desde el entorno de pruebas hasta la plataforma real de vuelo.

#### **4.3.1 FarsiteSim**

FarsiteSim es un servicio que al ejecutarlo abre el programa Farsite donde se puede observar la simulación del incendio con el que se ha trabajado.

Para ello ejecuta un fichero nombrado “Bookmark” en formato BMK que se ha generado anteriormente a través del simulador Farsite una vez introducidos todos los datos de entrada necesarios para la simulación.

En este fichero se encuentran almacenados de forma fija todos los rasters y parámetros de entrada que Farsite necesita para funcionar. De esta manera al ejecutar el “Bookmak” se abrirá el simulador Farsite donde simplemente se tendrán que seleccionar los ficheros que se deseen de salida y ya estará listo para que la simulación se inicialice.

#### **4.3.2 IRService**

Como bien se ha comentado en la introducción, en este proyecto se ha realizado un programa en lenguaje C# llamado IRService. Éste programa es el responsable de recoger los datos generados por el simulador Farsite y el simulador Flight Gear de tal manera que con estos datos el programa pueda informar de la temperatura que tienen cada uno de los puntos por donde va pasando el UAV.

Para entender mejor el funcionamiento del programa a continuación se van a explicar los puntos claves del mismo.

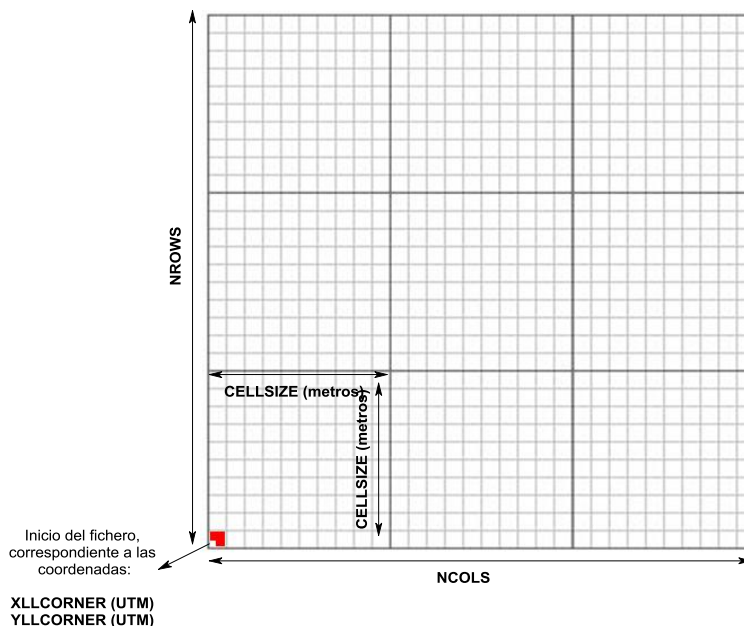
Antes de nada, hace falta comentar que este programa trabaja con los ficheros que crea el simulador Farsite, *Time of arrival* el cual aporta la información en formato horas y *Heat/Area* que aporta la información en formato  $\text{KJ/m}^2$ .

Dichos ficheros tienen la misma cabecera, la cual se puede observar en la **Tabla 4.1**.

**Tabla 4.1.** Encabezado de los ficheros *Time of arrival* y *Heat/Area*.

```
NCOLS 4133
NROWS 3433
XLLCORNER 430762.39
YLLCORNER 4592270.16
CELLSIZE 30.000000
```

Dónde NCOLS y NROWS son los valores que indican el número de columnas y filas respectivamente que forman el fichero. Los valores de XLLCORNER y YLLCORNER son las coordenadas X e Y en formato UTM que indican el inicio del fichero, siendo dicho inicio el punto que se puede observar de color rojo en la **Fig.4.4**. Por último, el valor de la opción CELLSIZE es el valor en metros que indica el tamaño que tiene cada celda UTM.



**Fig. 4.4** Representación grafica del significado de la cabecera de los ficheros *Time of arrival* y *Heat/Area*.

El UAV envía las coordenadas en formato GPS que el programa obtiene mediante la operación que se puede observar en **Tabla 4.2**.

**Tabla 4.2.** Transformación de coordenadas GPS a UTM.

```

p = UAV.currentPositionInRadians;
WgsPoint p_gps = new WgsPoint((p.Latitude * 180) /
Math.PI, (p.Longitude * 180) / Math.PI, p.Altitude);

log.Info(string.Format("Position (UTM): lat " + p_gps.getUtmY() +
", lon" + p_gps.getUtmX()));

```

La variable  $p$  es la que contiene el valor de las coordenadas GPS que se reciben del avión. Pero como se necesitan tener las coordenadas en formato UTM para así poder utilizar los ficheros que genera Farsite se ha realizado una operación para pasar de coordenadas GPS a UTM, el resultado de dicha operación se queda guardado temporalmente en  $p\_gps.getUtmY$  para obtener la coordenada Y en formato UTM, y en  $p\_gps.getUtmX$  para obtener la de la X.

Una vez se tiene el valor de las coordenadas en formato UTM se tiene que encontrar la fila y columna en los ficheros *Time o arrival* y *Heat/Area* correspondiente a dicha coordenada. Para ello se realiza la función que se puede ver en la **Tabla 4.3**.

**Tabla 4.3.** Columna y fila de los ficheros generados por Farsite correspondientes a las coordenadas UTM.

```

columna = Convert.ToInt32(p_gps.getUtmX() - xLowLeft) / celda;
fila = Convert.ToInt32(p_gps.getUtmY() - yLowLeft) / celda;

```

Para encontrar tanto la columna como la fila en los ficheros generados por Farsite simplemente se tiene que restar el valor de la coordenada UTM ( $p\_gps.getUtmX$  en el caso de la coordenada X,  $p\_gps.getUtmY$  en el caso de la coordenada Y) y restar de dicho valor la coordenada que indica la posición UTM correspondiente al inicio del fichero, información que se ha conseguido almacenando las opciones XLLCORNER y YLLCORNER de la cabecera de dichos ficheros en las variables del programa :  $xLowLeft$  y  $yLowLeft$  respectivamente.

Realizando esta operación se obtienen los metros de distancia del punto actual al inicial del fichero, y como bien ya se ha comentado anteriormente las celdas UTM tiene un tamaño en concreto que en este caso es el valor de la variable *celda* que contiene el valor del campo CELLSIZE de la cabecera. Así que para encontrar la columna y fila exacta se tiene que dividir el resultado de dicha operación entre el valor de *celda*.



#### 4.1.1.1 Conversión de calor a temperatura

Una vez ya se conoce la fila y columna a la cual el programa tiene que dirigirse para recoger la información, tiene que ser modificada la información de la temperatura para que de esta manera el programa pudiera dar el valor de la temperatura en grados en vez de  $\text{KJ/m}^2$ , que como ya se ha comentado anteriormente es el formato en el cual se encuentra la información en el fichero Heat/Area. Para ello se han declarado dos constantes (ver **Tabla 4.4**):

**Tabla 4.4.** Declaración de las variables “MAX\_HPA” y “MAX\_TEMP”.

```
private static readonly int MAX_HPA = 99999;  
private static readonly int MAX_TEMP = 900;
```

Donde “MAX\_HPA” es el máximo valor de temperatura en  $\text{KJ/m}^2$  que se puede encontrar en el fichero *Heat/Area*, y “MAX\_TEMP” es el valor mínimo de temperatura en  $\text{KJ/m}^2$  que se puede encontrar en dicho fichero. Por tanto para que el programa de la información en grados en vez de en  $\text{KJ/m}^2$  se a realizado la siguiente operación (ver **Tabla 4.5**):

**Tabla 4.5.** Función para pasar de  $\text{KJ/m}^2$  a grados.

```
temp = temp * MAX_TEMP / MAX_HPA;
```

Donde “temp” es la variable de la temperatura de cada uno de los puntos del mapa en formato  $\text{KJ/m}^2$ .

Para que la variable “temp” pase a tener formato de grados, se obtiene el valor de la variable “temp” en formato  $\text{KJ/m}^2$  que es multiplicado por el valor mínimo de temperatura del fichero (MAX\_TEMP) y a su vez se divide entre el valor máximo de temperatura del fichero (MAX\_HPA). Mediante la aplicación de factores de conversión se obtiene el resultado de la temperatura en grados.

#### 4.1.1.2 Degradado de la temperatura por el tiempo

El fichero *Heat/Area* solo contiene el valor de la temperatura de cada uno de los puntos del mapa del momento exacto en que el incendio llega a ellos. Es

decir, contiene el valor máximo de temperatura de cada uno de los puntos. Pero como el fuego se va deteriorando a medida que pasa el tiempo, se ha realizado la fórmula que se puede observar en la **Tabla 4.6** para que a cada hora que pase se vaya disminuyendo dicho valor.

**Tabla 4.6.** Función que se realiza cuando la hora actual es mayor que la hora que contiene el fichero *Time of Arrival*.

```
if (hora > temps_fitxer){  
  
    int t_pas = hora - temps_fitxer;  
    int degradat = (t_pas * MAX_HPA / 24);  
    temp = heat_fitxer - degradat;  
  
    .....
```

La variable “hora” es la hora actual y la variable “temps\_fitxer” es la hora que contiene el fichero *Time of arrival*. Este fichero solo tiene registrada la hora del momento exacto en el que el fuego llega a cada punto. Por lo tanto se ha declarado una variable “t\_pas” que contiene la diferencia de tiempo entre la hora actual y el valor de la hora de cada punto del fichero *Time of arrival*. Asumiendo que un mismo punto no puede estar quemándose durante más de 24 horas se ha realizado la fórmula del “degradat” que es la que hace que se disminuya el valor de la temperatura a cada hora que pasa.

La variable “degradat” es el resultado de la multiplicación de la variable “t\_pas” por la división del valor máximo de temperatura (MAX\_HPA) que puede tener el fichero *Heat/Area* entre las 24 horas que tiene un día. De esta manera se asegura que hasta el punto que tenga la máxima temperatura posible este apagado en un día.

Una vez se obtiene el valor de disminución de la temperatura del fuego (variable “degradat”), solo falta restar dicho valor de la variable “heat\_fitxer” (valor de temperatura que contiene el fichero *Heat/Area*). El resultado de la operación anterior será el nuevo valor de la variable “temp”.

Este último caso, solo sucede si la hora actual es mayor que el tiempo que contiene el fichero *Time of arrival* (se comprueba para a cada uno de los puntos del fichero).

En el caso en que la hora actual sea menor que el tiempo que contiene el fichero comentado anteriormente, quiere decir que el fuego aun no ha llegado a dicho punto y por tanto el valor de la temperatura en ese punto será 15° ya que es el valor aproximado que tiene la tierra en su estado natural (ver **Tabla 4.7**).

En el caso de que los dos valores de tiempo (el actual y el del fichero) sean iguales, el valor de la temperatura pasara a ser el mismo que el que hay en el fichero *Heat/Area* correspondiente a dicho punto (ver **Tabla 4.8**).

**Tabla 4.7.** Función que se realiza cuando la hora actual es menor que la hora que contiene el fichero *Time of Arrival*.

```
else if (hora < temps_fitxer){  
    temp = 15;  
}
```

**Tabla 4.8.** Función que se realiza cuando la hora actual es igual que la hora que contiene el fichero *Time of Arrival*.

```
else if (hora == temps_fitxer){  
    temp = heat_fitxer;  
    .....
```

#### 4.3.3 Fire Perimeter Visor

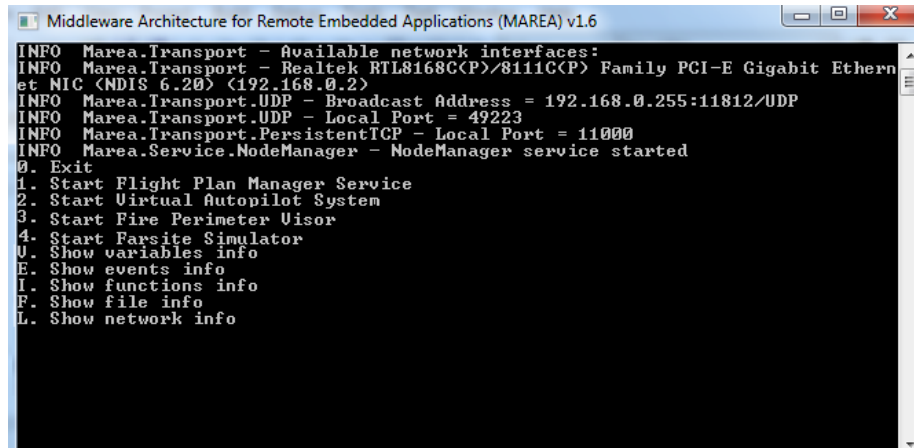
Fire Perimeter Visor es un servicio del sistema que obtiene a través de la salida del simulador Farsite los valores de las coordenadas que representan el perímetro del fuego. Con ellas genera un fichero KLM que permite visualizar el incendio en el programa Google Earth.

Las coordenadas se encuentran en formato UTM, pero Google Earth solo permite visualizar coordenadas en formato GPS, es por esto que antes de realizar el KLM el servicio realiza una transformación para pasar de UTM a GPS.

#### 4.4 Demostración

Una vez se tienen todos los servicios preparados y dispuestos para funcionar se han realizado las pruebas del funcionamiento de todo el sistema. El sistema está compuesto por los elementos que se han podido ver anteriormente en la **Fig.4.3**.

En el PC1 cuando se inicia la simulación aparece en la pantalla el menú de la **Fig.4.5**, donde se pueden observar los diferentes servicios disponibles en dicho ordenador. Para realizar esta prueba solo hace falta la opción 1 (Start Flight Plan Manager Service) ,2 (Start Virtual Autopilot System) y 3 (Start Fire Perimeter Visor).

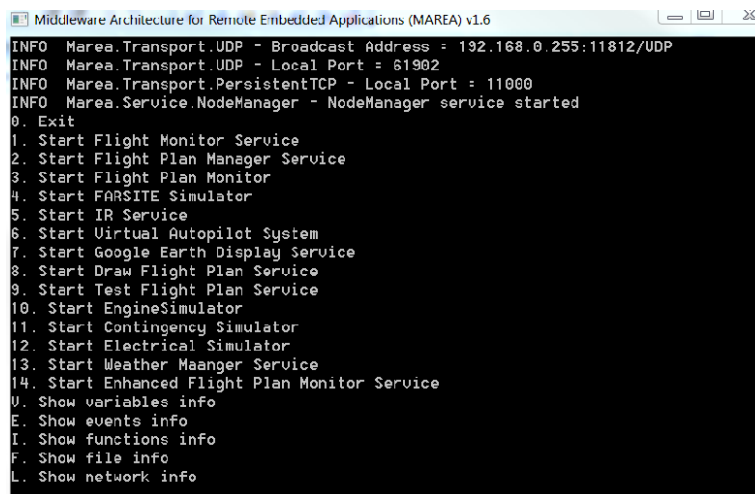


```

Middleware Architecture for Remote Embedded Applications (MAREA) v1.6
INFO Marea.Transport - Available network interfaces:
INFO Marea.Transport - Realtek RTL8168C(P)/8111C(P) Family PCI-E Gigabit Ethernet NIC (NDIS 6.20) (192.168.0.2)
INFO Marea.Transport.UDP - Broadcast Address = 192.168.0.255:11812/UDP
INFO Marea.Transport.UDP - Local Port = 49223
INFO Marea.Transport.PersistentTCP - Local Port = 11000
INFO Marea.Service.NodeManager - NodeManager service started
0. Exit
1. Start Flight Plan Manager Service
2. Start Virtual Autopilot System
3. Start Fire Perimeter Visor
4. Start Farsite Simulator
U. Show variables info
E. Show events info
I. Show functions info
F. Show file info
L. Show network info
  
```

**Fig. 4.5** Menú principal de los servicios ofrecidos en el PC1

Por otro lado en el PC2 también aparece en pantalla el menú correspondiente a los servicios que él puede ofrecer. Dicho menú se puede observar en la **Fig.4.6**. En este ordenador es donde se pondrán en funcionamiento los servicios que corresponden a los números 1 (Start Flight Monitor Service),3 (Strart Flight Plan Monitor), 4 (Start FARSITE Simulator), 5 (Start IR Service) y 7 (Strart Google Earth Display) del menú principal.



```

Middleware Architecture for Remote Embedded Applications (MAREA) v1.6
INFO Marea.Transport.UDP - Broadcast Address = 192.168.0.255:11812/UDP
INFO Marea.Transport.UDP - Local Port = 61902
INFO Marea.Transport.PersistentTCP - Local Port = 11000
INFO Marea.Service.NodeManager - NodeManager service started
0. Exit
1. Start Flight Monitor Service
2. Start Flight Plan Manager Service
3. Start Flight Plan Monitor
4. Start FARSITE Simulator
5. Start IR Service
6. Start Virtual Autopilot System
7. Start Google Earth Display Service
8. Start Draw Flight Plan Service
9. Start Test Flight Plan Service
10. Start EngineSimulator
11. Start Contingency Simulator
12. Start Electrical Simulator
13. Start Weather Manager Service
14. Start Enhanced Flight Plan Monitor Service
U. Show variables info
E. Show events info
I. Show functions info
F. Show file info
L. Show network info
  
```

**Fig. 4.6** Menú principal de los servicios ofrecidos en el PC2.

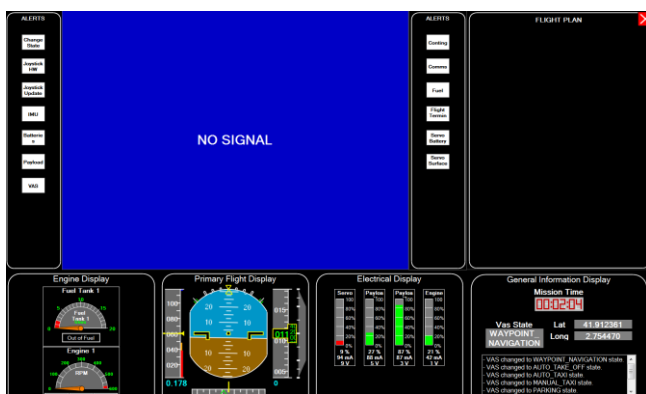
Pero antes de seleccionar los servicios correspondientes, se tiene que asegurar que la simulación se está ejecutando en los dos ordenadores a la vez,

ya que los servicios de los dos ordenadores interactúan entre ellos y si en un ordenador no se está ejecutando la simulación repercutiría en el funcionamiento de los servicios del otro ordenador.

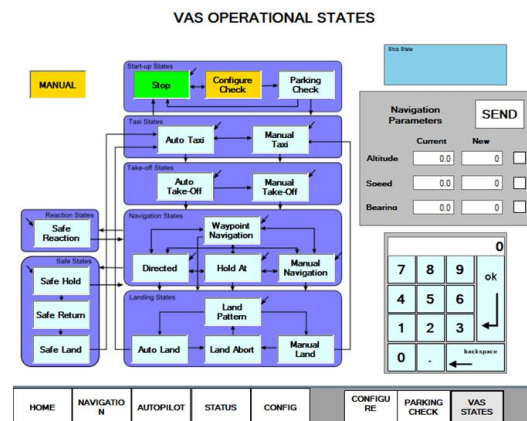
Una vez se ha comprobado la ejecución de la simulación en los dos ordenadores, se pasarán a activar los servicios de cada ordenador que se han comentado anteriormente.

En el PC1, una vez seleccionados los servicios aparecerá en pantalla la ventana del Virtual Autopilot System .

Por otro lado cuando en el PC2 se seleccionen los servicios correspondientes aparecerá en la doble pantalla la ventana del servicio Flight Monitor Service (ver **Fig.4.7**), la del Flight Plan Monitor y el simulador Farsite con la simulación del incendio.



a) Pantalla 1



b) Pantalla 2

**Fig. 4.7** Flight Monitor Service.

Una vez abierto el Flight Monitor Service, se tiene que ir pasando manualmente de estados, tal y como se puede observar en la **Fig.4.7** apartado b). Al pasar del estado “Stop” al “Configure check”, este envía un orden al PC1 y hace que se ejecute el simulador Flight Gear (ver **Fig.4.8**).

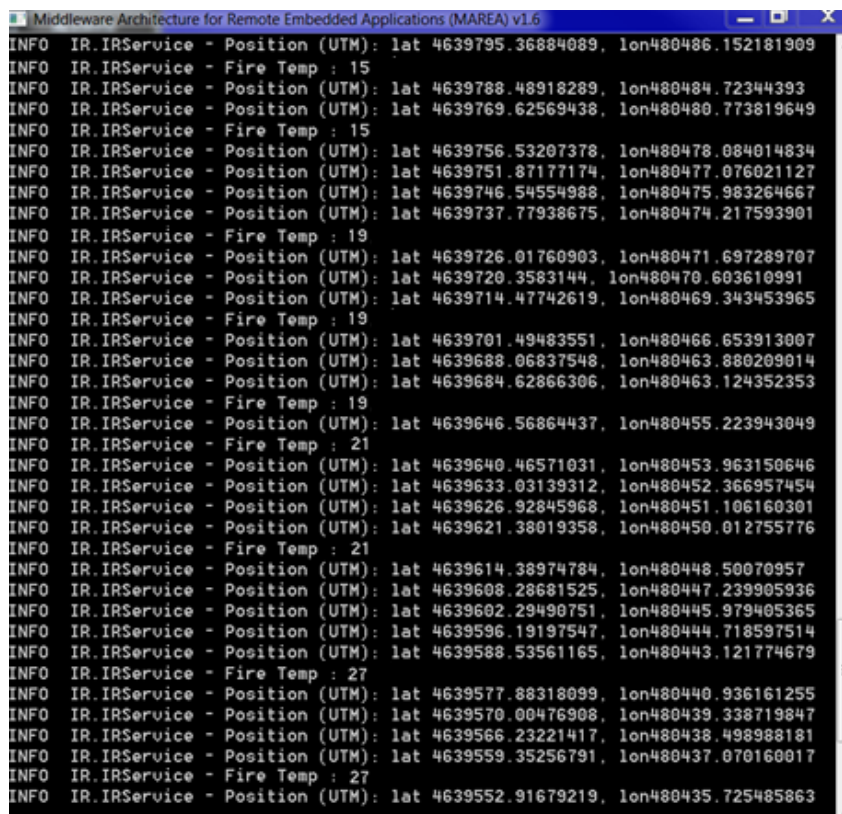


**Fig. 4.8** Simulador Flight Gear.

Seguidamente se tiene que seleccionar el plan de vuelo que tendrá que seguir el simulador Fligh Gear a través de la ventana del Flight Plan.

Después de enviar el Plan de vuelo se tienen que ir cambiando los estados del servicio Flight Monitor Service para finalmente llegar al estado “Waipoint Navigation” donde a partir de aquí el Flight Gear automáticamente realizará el recorrido deseado.

Una vez el avión este volando el servicio IRService imprimirá por pantalla las coordenadas por donde va pasando el avión junto con la temperatura que hay en dicha coordenada, tal y como se puede observar en la **Fig.4.9**.



```

Middleware Architecture for Remote Embedded Applications (MARE) v1.6
INFO IR.IRService - Position (UTM): lat 4639795.36884089, lon480486.152181909
INFO IR.IRService - Fire Temp : 15
INFO IR.IRService - Position (UTM): lat 4639788.48918289, lon480484.72344393
INFO IR.IRService - Position (UTM): lat 4639769.62569438, lon480480.773819649
INFO IR.IRService - Fire Temp : 15
INFO IR.IRService - Position (UTM): lat 4639756.53207378, lon480478.084014834
INFO IR.IRService - Position (UTM): lat 4639751.87177174, lon480477.076021127
INFO IR.IRService - Position (UTM): lat 4639746.54554988, lon480475.983264667
INFO IR.IRService - Position (UTM): lat 4639737.77938675, lon480474.217593901
INFO IR.IRService - Fire Temp : 19
INFO IR.IRService - Position (UTM): lat 4639726.01760903, lon480471.697289707
INFO IR.IRService - Position (UTM): lat 4639720.3583144, lon480470.603610991
INFO IR.IRService - Position (UTM): lat 4639714.47742619, lon480469.343453965
INFO IR.IRService - Fire Temp : 19
INFO IR.IRService - Position (UTM): lat 4639701.49483551, lon480466.653913007
INFO IR.IRService - Position (UTM): lat 4639688.06837548, lon480463.880209014
INFO IR.IRService - Position (UTM): lat 4639684.62866306, lon480463.124352353
INFO IR.IRService - Fire Temp : 19
INFO IR.IRService - Position (UTM): lat 4639646.56864437, lon480455.223943049
INFO IR.IRService - Fire Temp : 21
INFO IR.IRService - Position (UTM): lat 4639640.46571031, lon480453.963150646
INFO IR.IRService - Position (UTM): lat 4639633.03139312, lon480452.366957454
INFO IR.IRService - Position (UTM): lat 4639626.92845968, lon480451.106160301
INFO IR.IRService - Position (UTM): lat 4639621.38019358, lon480450.012755776
INFO IR.IRService - Fire Temp : 21
INFO IR.IRService - Position (UTM): lat 4639614.38974784, lon480448.50070957
INFO IR.IRService - Position (UTM): lat 4639608.28681525, lon480447.239905936
INFO IR.IRService - Position (UTM): lat 4639602.29490751, lon480445.979405365
INFO IR.IRService - Position (UTM): lat 4639596.19197547, lon480444.718597514
INFO IR.IRService - Position (UTM): lat 4639588.53561165, lon480443.121774679
INFO IR.IRService - Fire Temp : 27
INFO IR.IRService - Position (UTM): lat 4639577.88318099, lon480440.936161255
INFO IR.IRService - Position (UTM): lat 4639570.00476908, lon480439.338719847
INFO IR.IRService - Position (UTM): lat 4639566.23221417, lon480438.498988181
INFO IR.IRService - Position (UTM): lat 4639559.35256791, lon480437.070160017
INFO IR.IRService - Fire Temp : 27
INFO IR.IRService - Position (UTM): lat 4639552.91679219, lon480435.725485863
  
```

**Fig. 4.9** IR Service.

Cuando el avión termine de realizar su recorrido volverá al de nuevo al punto de inicio, donde aterrizará.

## CAPÍTULO 5. CONCLUSIONES

### 5.1 Conclusiones Tecnológicas

Como conclusiones tecnológicas se puede afirmar que se ha conseguido el objetivo final de este proyecto.

Para poderlo realizar primero se ha tenido que estudiar el simulador de incendios Farsite, escogido debido a tratarse de un programa de software libre, con lo cual si durante el avance del proyecto se hubiera tenido que modificar el código para adecuarse más a los objetivos finales, no hubiera habido ningún tipo de problema. Pero gracias al profundo estudio realizado no se ha tenido que modificar el código del programa y se ha consiguiendo extraer la información necesaria.

Al principio hubieron problemas para poder encontrar los ficheros necesarios que se necesitaban para poder realizar la simulación en el terreno que se deseaba (zona de Catalunya). Finalmente y como consecuencia de la investigación realizada se consiguieron generar de forma coherente los ficheros necesarios solucionando así el problema.

Después de terminar con el estudio del simulador Farsite y poder simular la zona deseada, se hicieron un conjunto de pruebas para comprobar que su funcionamiento se había comprendido correctamente y realmente el simulador realizaba la acción deseada. De dichas pruebas se obtuvieron los resultados esperados.

Una vez obtenidos los resultados de las diferentes simulaciones, se implementó el programa en código C# descrito en el proyecto. Se escogió realizar el programa en código C# porque este programa es un servicio adicional a un proyecto que realiza el equipo ICARUS de la EPSC y el código utilizado para dicho proyecto se encuentra en C#, de esta manera se han podido utilizar clases y librerías ya creadas, con lo cual ha sido en cierta manera un ahorro de tiempo.

Lo más complicado que se ha encontrado al realizar el código del programa es el cómo tener que sincronizar la simulación del incendio con la simulación del avión, ya que el avión utiliza como referencia las coordenadas GPS y el simulador Farsite utiliza un fichero organizado en filas y columnas, así que se tuvo que encontrar la manera de corresponder dichas filas y columnas con las coordenadas GPS del avión y de esta manera saber realmente cuando pasaba el avión por encima del fuego. Finalmente se consiguió dicho objetivo de forma satisfactoria.

Por último cuando se realizaron las pruebas de todo el sistema funcionando conjuntamente hubo algunos problemas de sincronización entre ellos, que finalmente se pudieron resolver fácilmente.

## 5.2 Conclusiones personales

Personalmente estoy satisfecha con el trabajo realizado en este proyecto, creo que me ha ayudado a terminar de formarme como ingeniera.

Gracias a la realización de este trabajo he aprendido a investigar, resolver problemas de forma eficaz, consolidar conocimientos de programación y poner en práctica algunos de los conceptos aprendidos en la carrera.

Al tener que utilizar un simulador de incendios y uno de vuelo he tenido que investigar en un campo para mi totalmente desconocido como es el de la prevención y extinción de incendios y los diferentes tipos de SIG y coordenadas geográficas que se utilizan actualmente.

Como consecuencia de la investigación realizada me ha hecho ver que los factores a tener en cuenta en el comportamiento del fuego son muchos más de los que me imaginaba.

También he aprendido a como trabajar y organizarme de forma eficaz, ya que la elaboración de este proyecto se ha ido haciendo por etapas, comprobando primeramente el funcionamiento de cada uno de los elementos que forman parte del sistema y terminando con la simulación de todo el conjunto, realizando paralelamente la memoria del trabajo en cada una de las diferentes etapas.

Finalmente solo hace falta añadir que la realización de este trabajo ha sido realmente satisfactoria.

## 5.3 Futuras líneas de trabajo

Este sistema presenta varias líneas futuras de trabajo todavía pendientes de crear.

En este TFC se ha presentado un sistema que en términos generales el objetivo final es que avisara mediante la pantalla del ordenador de los puntos calientes que existen en un incendio, por esto una continuación posible es generar con esta información un mapa donde se visualizaran dichos puntos, así su localización quedaría aun más clara.

En capítulo 3 de este TFC se ha podido ver el trabajo costoso que se ha tenido que hacer para poder generar los rasters necesarios que se necesitan para poder llevar a cabo la simulación. Por esto, otra línea de trabajo futura podría ser la creación de un servidor donde se encontraran en él todos los rasters necesarios en el formato aceptado para Farsite y de esta manera poder simular un incendio en cualquier lugar del que se desee.

Por último una línea de trabajo futura y necesaria sería la de realizar todo este TFC en un entorno real, es decir, utilizando un UAV real, con una cámara térmica en el interior y ejecutar los servicios trabajados en este proyecto.



## **5.4 Impacto Medioambiental**

Como bien se ha comentado en la introducción, los incendios forestales provocan un alto impacto ambiental económico y social.

Los incendios destruyen ecosistemas que provocan a su vez la destrucción de las especies que habitan en él, repercutiendo así en un impacto visual degradado. Éste hecho también afecta a la economía y a la sociedad de hoy en día ya que muchas personas pierden sus tierras y trabajos. También son muchas las vidas humanas que se han perdido por intentar evitar que el fuego se propague o encontrarse inesperadamente rodeados en llamas.

Este TFC quiere ser una herramienta más para ayudar a las personas que trabajan en la extinción de los incendios y de esta manera que puedan ser más eficientes en su trabajo y puedan apagar los incendios con más fiabilidad y rapidez. Es por esto que el impacto medioambiental es realmente positivo.

## BIBLIOGRAFIA

- [1] MA Finney. *Calculation of FIRE spread rates across random landscapes*. Internacional Journal of Wildland FIRE, 2003.
- [2] Simulador Flight Gear: <http://www.flightgear.org/>
- [3] Ortega Parrado S., TFC: *Red-Eye. Sistema para la monitorización semiautomática de incendios forestales*, EPSC: 9 de junio 2009.
- [4] Gutiérrez Puebla J.; Gould M., *SIG: Sistemas de Información Geográfica*, Madrid: Síntesi, DL 1994.
- [5] Enciclopedia libre Wikipedia: "Sistema de información Geográfica". Última consulta: 15 de Marzo 2010.  
[http://es.wikipedia.org/wiki/Sistema\\_de\\_Informaci%C3%B3n\\_Geogr%C3%A1fica](http://es.wikipedia.org/wiki/Sistema_de_Informaci%C3%B3n_Geogr%C3%A1fica)
- [6] Otero Pastor I. *.Paisaje, Teledetección y SIG: Conceptos y aplicaciones*, Madrid: Fundación Conde del Valle de Salazar, DL 1999.
- [7] "Sistemas de Información geográfica". Última consulta: 15 de Marzo 2010  
<http://exp-grafica.uma.es/Profesores/www-jrad/document/gis/sig.pdf>
- [8] "El sistema de posicionamiento global y los receptores GPS". Última consulta : 15 de Marzo 2010. <http://www.efdeportes.com/efd9/gps.htm>
- [9] Blanco Lopez J., "GPS: Tecnología más allá de la atmosfera". Última consulta:15 de Marzo 2010.
- [10] Enciclopedia libre Wikipedia : "Proyección cartográfica". Última consulta: 9 de Abril 2010.  
[http://es.wikipedia.org/wiki/Proyecci%C3%B3n\\_cartogr%C3%A1fica](http://es.wikipedia.org/wiki/Proyecci%C3%B3n_cartogr%C3%A1fica)
- [11] Enciclopedia libre Wikipedia: "Sistema de Coordenadas Universal Transversal Mercator". Última consulta: 12 de Abril 2010

[http://es.wikipedia.org/wiki/Sistema\\_de\\_Coordenadas\\_Universal\\_Transversal\\_de\\_Mercator](http://es.wikipedia.org/wiki/Sistema_de_Coordenadas_Universal_Transversal_de_Mercator)

- [12] Fernandez-Coppel I.A, “Las Coordenadas Geográficas y la proyección UTM”, Universidad de Valladolid. 13 de Febrero del 2001.
- [13] “Lambert azimuthal equal-area projection”. Última consulta: 10 de Marzo 2010.  
[http://en.wikipedia.org/wiki/Lambert\\_azimuthal\\_equal-area\\_projection](http://en.wikipedia.org/wiki/Lambert_azimuthal_equal-area_projection)
- [14] “Ejemplos gráficos de las proyecciones geográficas más comunes”. Última consulta 16 Marzo 2010.  
<http://www.elgps.com/documentos/proyecciones/proyecciones.html#LAEA>
- [15] Galán Santano M.; Molina Terrén D., “Plan de Gestion de Grandes Incendios Forestales en e Berguedà”, DARP\_DA a la UdL – 1998-2000.
- [16] Martínez López E.; Catellnou Ribau M.; Molina Terrén D., “Pre-extinció de grandes Incendios forestales: El caso de Rialb”, Universidad Autònoma y Universidad de Lleida.
- [17] Castrillón M.; Jorge P.A.; López I.J.; Macías A.; Martín D.; Nebot R.J.; Sabbagh I.; Sánchez J.; Sánchez A.J.; Suárez J.P.; Trujillo A., “Aplicación para la gestión de emergencias en la comunidad canaria: el caso de predicción y simulación de incendios forestales” , Universidad de las Palmas de Gran Canaria.
- [18] Simulation Flow diagram. Última consulta 27 de Febrero 2010.  
<http://proceedings.esri.com/library/userconf/proc95/to100/p092.html>
- [19] García E.M; Serna M.A.; Bermúdez A.; Casado R., “Simulación de un sistema de apoyo en la extinció de incendios forestales basado en WSN”, Instituto de Investigación en Informática de Albacete, Departamento de Sistemas Informáticos, Universidad de Castilla-La Mancha.
- [20] Arnaldos Viger J.; Navalón Nonell X.; Pastor Ferrer E.; Planas Cuchi E.; Zárate López L., *Manual de ingeniería básica para la prevención y extinció de incendios forestales*, Madrid: Mundi-Prensa, 2003.
- [21] Modelos de combustible de NFFL :

<http://www.fs.fed.us/r5/rsl/projects/frdb/tables/table131.html>

[22] Escena de montaña : <http://en.fotolia.com/id/10278811>

[23] ArcView: <http://www.esri.com/software/arcview/index.html>

[24] USGS:  
[http://eros.usgs.gov/#/Find\\_Data/Products\\_and\\_Data\\_Available/gtopo30/hydro](http://eros.usgs.gov/#/Find_Data/Products_and_Data_Available/gtopo30/hydro)

[25] El programa Global Mapper: <http://www.globalmapper.com> (hay la posibilidad de descargarse una demo de forma gratuita o bien comprarlo).

[26] CREAM: <http://www.cream.uab.es/iefc/pub/Regions/PortadaRF3.htm>

[27] Institut Cartogràfic de Catalunya :  
<http://draco.icc.cat/web/content/php/geotex/geoutm.php>

[28] Ricard Morales P. "TFC: Plataforma d'orientació pel seguiment en temps real d'UAVs", Technical University of Catalonia, 3 de Novembre de 2009.

[29] López J.; Royo P.; Pastor E.; Barrado C.; Santamaria E., "A Middleware Architecture for Unmanned Aircraft Avionics", Technical University of Catalonia.

[30] Royo P.; Lopez J.; Tristanchó J.; Lema J.M; Lopez B.; Pastor E.; "Service Oriented Fast Prototyping Environment for UAS Missions", Computer Architecture Dept., UPC, Spain

[31] Google Earth : [http://earth.google.es/intl/es\\_es/download-earth.html](http://earth.google.es/intl/es_es/download-earth.html)



**Escola Politècnica Superior  
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# ANNEXOS

**TITULO DEL TFC:** Integración de Farsite en el entorno de simulación para UAS.

**TITULACIÓN:** Ingeniería Técnica de Telecomunicación, especialidad Telemática

**AUTOR:** M<sup>a</sup> Victoria Noguero Planas

**DIRECTOR:** Cristina Barrado Muxí

**FECHA:** 12 de Mayo de 2010

A continuación se puede observar la tabla de combustibles utilizada para realizar el raster del fuel. ([www.floresta.ufpr.br/firelab/artigos/artigo358.pdf](http://www.floresta.ufpr.br/firelab/artigos/artigo358.pdf))

### 1.1. Anexo 1: Modelos de combustible propuestos por Rothermel para el NFDRS y adaptados por la DGCN a España.

Grupo	Modelo	Descripción
<b>PASTOS</b>	1	Pasto fino, seco y bajo, que recubre completamente el suelo. El matorral o el arbolado cubren menos de un tercio de la superficie. El fuego se propaga rápidamente por el pasto seco.
	2	Pastizal con presencia de matorral o arbolado claro que cubren entre uno y dos tercios de la superficie. El combustible está formado por el pasto seco, la hojarasca y ramillas caídas de la vegetación leñosa. El fuego corre rápidamente por el pasto seco.
	3	Pastizal espeso y alto (> 1m.). Es el modelo típico de las sabanas. Los campos de cereales son representativos de este modelo. Los incendios son los más rápidos y de mayor intensidad.
<b>MATORRAL</b>	4	Matorral o arbolado joven muy denso de unos 2 m de altura. Continuidad horizontal y vertical del combustible. Abundancia de combustible leñoso muerto (ramas) sobre plantas vivas. El fuego se propaga rápidamente sobre las copas del matorral con gran intensidad y llamas grandes. La humedad del combustible vivo tiene gran influencia en el comportamiento del fuego.
	5	Matorral denso pero bajo, de altura no superior a 0,6 m. Cargas ligeras de hojarasca del mismo matorral, que contribuye a propagar el fuego con vientos flojos. Fuegos de intensidad moderada.
	6	Matorral más viejo que en el modelo 5, con alturas entre 0,6 y 1,2 m. Los combustibles vivos son más escasos y dispersos. El conjunto es más inflamable que el modelo 5. El fuego se propaga a través del matorral con vientos de moderados a fuertes.  Matorral inflamable de 0,6 a 2,0 m de altura, que propaga el fuego bajo el arbolado. El incendio se

	7	desarrolla con contenidos más altos en humedad del combustible muerto que en los otros modelos debido a la naturaleza más inflamable de los combustibles vivos.
<b>HOJARRASCA BAJO ARBOLADO</b>	8	Hojarasca en bosque denso de coníferas o frondosas. La hojarasca forma una capa compacta al estar formada por acículas cortas (5 cm o menos) o por hojas planas no muy grandes. Los fuegos son de poca intensidad, con llamas cortas y velocidades de avance bajas. Solamente en condiciones meteorológicas desfavorables (altas temperaturas, bajas humedades relativas y vientos fuertes) este modelo puede volverse peligroso.
	9	Hojarasca en bosque denso de coníferas o frondosas, que se diferencia del modelo 8 en que forma una capa esponjada poco compacta, con mucho aire interpuesto. Está formada por acículas largas, como en masas de Pinus pinaster, o por hojas grandes y rizadas como las de Quercus pyrenaica, Castanea sativa, etc. Los fuegos son más rápidos y con llamas más largas que en el modelo 8.
<b>RESTOS DE CORTA Y OPERACIONES SELVÍCOLAS</b>	10	Restos leñosos originados naturalmente, incluyendo leña gruesa caída como consecuencia de vendavales, plagas intensas, o excesiva madurez de la masa, con presencia de vegetación herbácea y matorral que crece entre los restos leñosos.
	11	Restos ligeros ( $\varnothing < 7,5$ cm) recientes, de tratamientos selvícolas o de aprovechamientos, formando una capa poco compacta de escasa altura (alrededor de 30 cm). La hojarasca y el matorral presentes ayudarán a la propagación del fuego. Los incendios tendrán intensidades altas y pueden generar pavesas.
	12	Restos más pesados que en el modelo 11, formando una capa continua de mayor altura (hasta 60 cm). Más de la mitad de las hojas están aún adheridas a las ramas sin haberse secado completamente. No hay combustibles vivos que influyan en el fuego. Los incendios tendrán intensidades altas y pueden generar pavesas.
	13	Grandes acumulaciones de restos gruesos ( $\varnothing > 7,5$ cm) y pesados, cubriendo todo el suelo.

A continuación se puede ver el Script que se ha realizado para modificar los rasters slope y aspect y para generar los rasters canopy y fuel.

## 1.2. Anexo 2: Script

```
echo "Saca los puntos y decimales del demCat9 y los pone en el fichero
demCat (sin tocar la cabecera)"
sed 7,/^END/{'s/\. [0-9][0-9]//g'} demCat9.asc>demCat.asc

echo "Empieza la transformación de demCat2.asc a canopyCat.asc"
sed 7,/^END/{'s/\<[0-9]\>/31/g'} demCat.asc|
sed 7,/^END/{'s/\<[0-9][0-9]\>/31/g'} |
sed 7,/^END/{'s/\<1[0-9][0-9]\>/31/g'} |
sed 7,/^END/{'s/\<[2-3][0-9][0-9]\>/13/g'} |
sed 7,/^END/{'s/\<[4-5][0-9][0-9]\>/14/g'} |
sed 7,/^END/{'s/\<[6-7][0-9][0-9]\>/15/g'} |
sed 7,/^END/{'s/\<[8-9][0-9][0-9]\>/11/g'} |
sed 7,/^END/{'s/\<1[0-1][0-9][0-9]\>/8/g'} |
sed 7,/^END/{'s/\<1[2-3][0-9][0-9]\>/5/g'} |
sed 7,/^END/{'s/\<1[4-9][0-9][0-9]\>/4/g'} |
sed 7,/^END/{'s/\<2[0-1][0-9][0-9]\>/4/g'} |
sed 7,/^END/{'s/\<2[2-3][0-9][0-9]\>/3/g'} |
sed 7,/^END/{'s/\<2[4-5][0-9][0-9]\>/2/g'} |
sed 7,/^END/{'s/\<2[6-9][0-9][0-9]\>/1/g'} >canopyCat.asc

echo "El archivo canopyFinal.asc se ha generado correctamente"

echo "Inicio de la transformación de demCat.asc a fuelCat.asc"

sed 7,/^END/{'s/\<[0-9]\>/6/g'} demCat.asc|
sed 7,/^END/{'s/\<[0-9][0-9]\>/6/g'} |
sed 7,/^END/{'s/\<1[0-9][0-9]\>/6/g'} |
sed 7,/^END/{'s/\<[2-3][0-9][0-9]\>/6/g'} |
sed 7,/^END/{'s/\<[4-5][0-9][0-9]\>/4/g'} |
sed 7,/^END/{'s/\<[6-7][0-9][0-9]\>/8/g'} |
sed 7,/^END/{'s/\<[8-9][0-9][0-9]\>/9/g'} |
sed 7,/^END/{'s/\<1[0-1][0-9][0-9]\>/10/g'} |
sed 7,/^END/{'s/\<1[2-3][0-9][0-9]\>/7/g'} |
sed 7,/^END/{'s/\<1[4-9][0-9][0-9]\>/5/g'} |
sed 7,/^END/{'s/\<2[0-1][0-9][0-9]\>/5/g'} |
sed 7,/^END/{'s/\<2[2-3][0-9][0-9]\>/2/g'} |
sed 7,/^END/{'s/\<2[4-5][0-9][0-9]\>/0/g'} |
sed 7,/^END/{'s/\<2[6-9][0-9][0-9]\>/0/g'} >fuelCat.asc

echo "El archivo fuelCat.asc se ha generado correctamente"

echo "empieza transformación del slopeCat9 a slopeCat"

sed 7,/^END/{'s/\. [0-9][0-9]//g'} slopeCat9.asc|
sed 7,/^END/{'s/\<[0-9][0-9]\>/0.&&/g'} |
sed 7,/^END/{'s/-9999/HOLA/g'} |
sed 7,/^END/{'s/[0-9][0-9]\>//g'} |
sed 7,/^END/{'s/HOLA/-9999/g'} >slopeCat.asc

echo "El archivo slopeCat.asc se ha generado correctamente"

echo "empieza transformación del aspectCat9 a aspectCat"
```



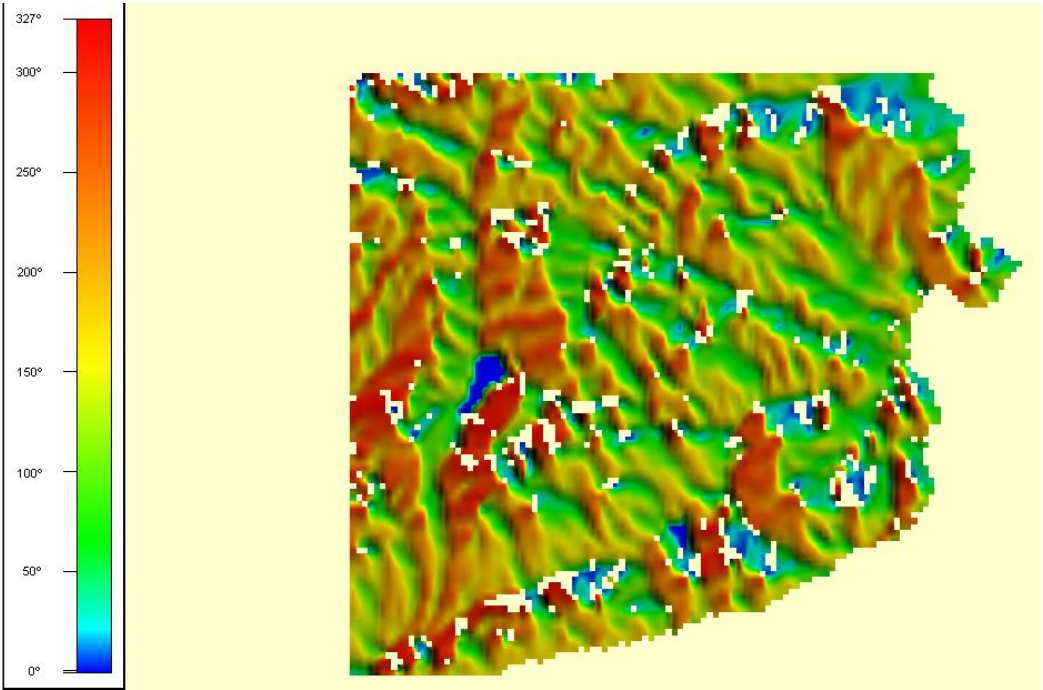
```
sed 7,/^END/{'s/\.[0-9][0-9]//g'} aspectCat9.asc|
sed 7,/^END/{'s/\<[0-9][0-9]\>/0.&&/g'} |
sed 7,/^END/{'s/-9999/HOLA/g'} |
sed 7,/^END/{'s/[0-9][0-9]\>//g'} |
sed 7,/^END/{'s/HOLA/-9999/g'} >aspectCat.asc
```

```
echo "El archivo aspectCat.asc se ha generado correctamente"
```

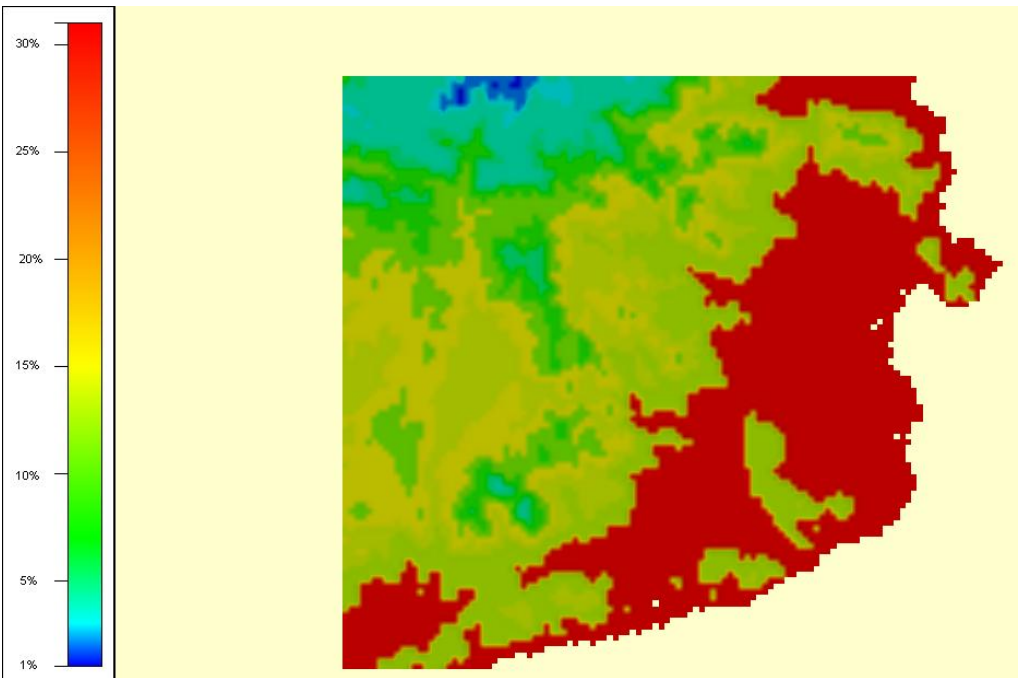
A continuación se muestran los rasters de entrada para la simulación de Farsite que se han utilizado en este proyecto.

### 1.3. Anexo 1: Rasters de entrada para Farsite

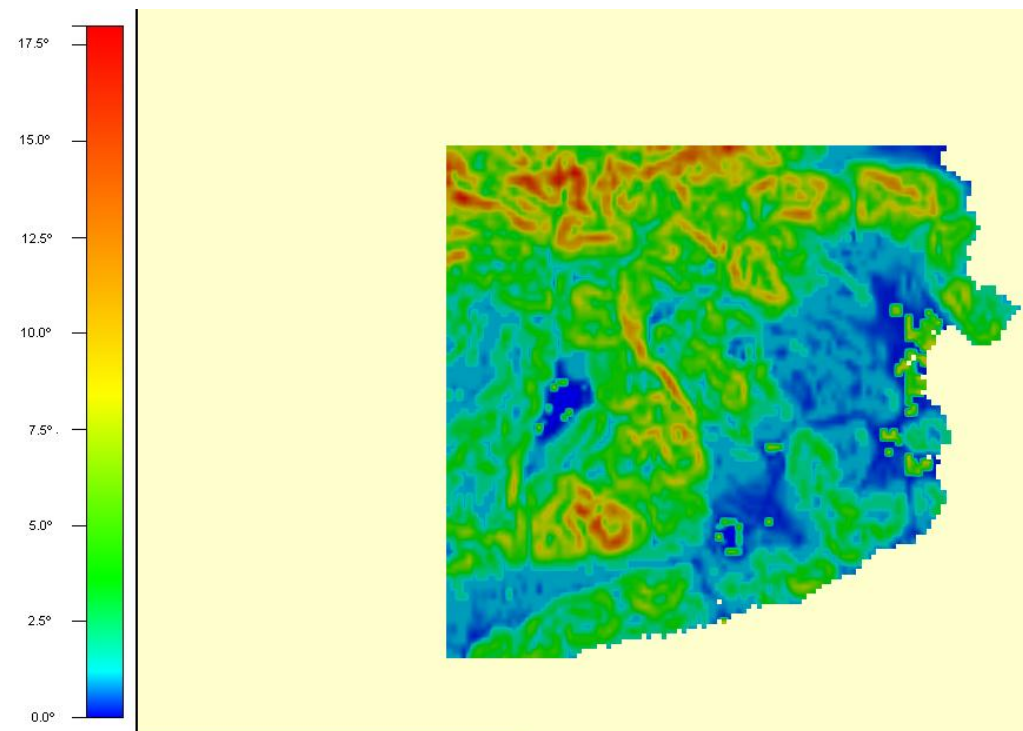
**Aspect**



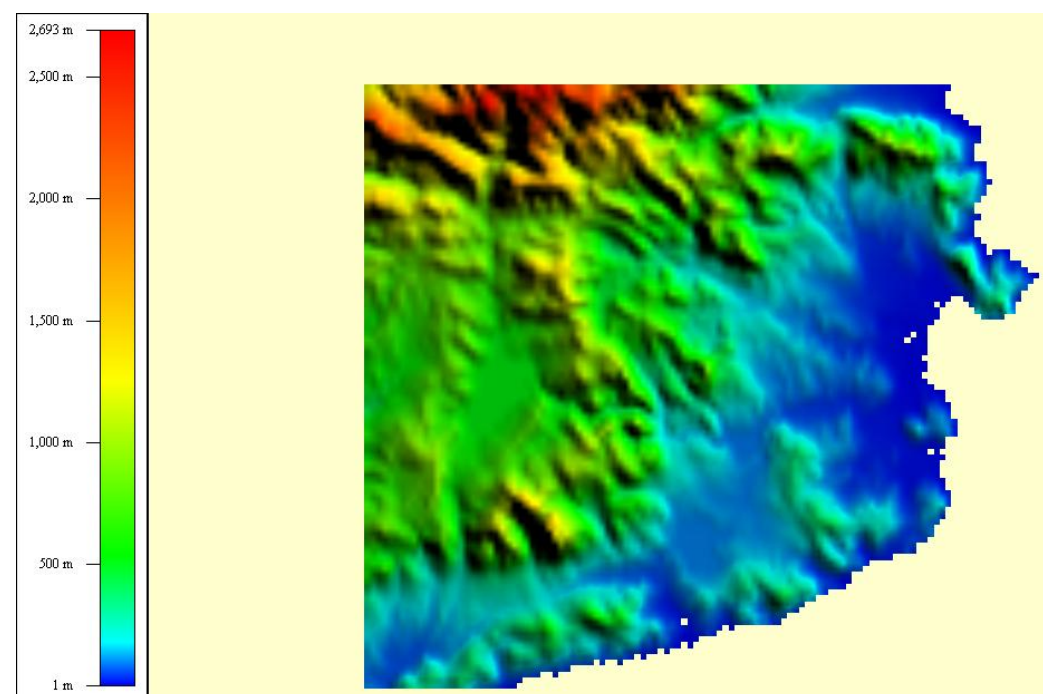
**Canopy**



## Slope



## Dem



## Fuel

